Consensus versus Expertise : A Case Study of Word Alignment with Mechanical Turk

Qin Gao and **Stephan Vogel**
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA, 15213
{qing,stephan.vogel}@cs.cmu.edu

Abstract

Word alignment is an important preprocessing step for machine translation. The project aims at incorporating manual alignments from Amazon Mechanical Turk (MTurk) to help improve word alignment quality. As a global crowdsourcing service, MTurk can provide flexible and abundant labor force and therefore reduce the cost of obtaining labels. An easy-to-use interface is developed to simplify the labeling process. We compare the alignment results by Turkers to that by experts, and incorporate the alignments in a semi-supervised word alignment tool to improve the quality of the labels. We also compared two pricing strategies for word alignment task. Experimental results show high precision of the alignments provided by Turkers and the semi-supervised approach achieved 0.5% absolute reduction on alignment error rate.

1 Introduction

Word alignment is used in various natural language processing tasks. Most state-of-the-art statistical machine translation systems rely on word alignment as a preprocessing step. The quality of word alignment is usually measured by AER, which is loosely related to BLEU score (Lopez and Resnik, 2006). There has been research on utilizing manually aligned corpus to assist automatic word alignment, and obtains encouraging results on alignment error rate. (Callison-Burch et al., 2004; Blunsom and Cohn, 2006; Fraser and Marcu, 2006; Niehues and Vogel, 2008; Taskar et al., 2005; Liu et al., 2005; Moore, 2005). However, how to obtain large amount of alignments with good quality is problematic. Labeling word-aligned parallel corpora requires significant amount of labor. In this paper we explore the possibility of using Amazon Mechanical Turk (MTurk) to obtain manual word alignment faster, cheaper, with high quality.

Crowdsourcing is a way of getting random labor force on-line with low cost. MTurk is one of the leading providers for crowdsourcing marketplace. There have been several research papers on using MTurk to help natural language processing tasks, Callison-Burch (2009) used MTurk to evaluate machine translation results. Kittur et al. (2008) showed the importance of validation data set, the task is evaluating quality of Wikipedia articles. There are also experiments use the annotation from MTurk in place of training data. For example (Kaisser et al., 2008) and (Kaisser and Lowe, 2008) used MTurk to build question answering datasets and choose summary lengths that suite the need of the users.

Word alignment is a relatively complicate task for inexperienced workers. The fact puts us in a dilemma, we can either provide lengthy instructions and train the workers, or we must face the problem that workers may have their own standards. The former solution is impractical in the context of crowdsourcing because heavily trained workers will expect higher payment, which defeats economical nature of crowdsourcing. Therefore we are forced to face the uncertainty, and ask ourselves the following questions: First, how consistent would the labels from random labelers be, given minimal or no instructions? Second, how consistent would these intuitive labels be consistent with the labels from expert labelers? Third, if there is certain level of consistency between the intuitive labels and the labels from experts, can we extract most reliable links from the former? Last but not least, given the alignment links, can we utilize them to help automatic word alignment without further human efforts?

The statistics on the data we get shows the internal consistency among multiple MTurk alignments is greater than 70%, and the precision is greater than 84% when consider all the links. By applying majority vote and consensus strategies, we can select links that have greater than 95% accuracy. When applying the alignment links on a new aligner that can perform constrained EM algorithm for IBM models we observe 0.5% absolute improvements on alignment error rate. The average per-word cost is about 2 cent per word.

The paper will be organized as follows, first we will discuss the design principle of the task and the implementation of the application for word alignment in section 2. Section 3 describes the algorithm used in utilizing the manual alignments. Section 4 presents the analysis on the harvested data and the expert labels, and the the experiment results of semi-supervised word alignment. Section 5 concludes the paper.

Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 30–34,
Los Angeles, California, June 2010.

2 Design of the task

In this task, we want to collect manual word alignment data from MTurk workers, Figure 2 shows an example of word alignment. There are two sentences which are translation of each other. There are links between words in two sentences, indicating the words are translation pairs. Notice that one word can be aligned to zero or more words, if a word is aligned to zero word, we can assume it is aligned to a virtual empty word. Therefore, given a sentence pair, we want workers to link words in source sentence to one or more target words or the empty word.

In our experiment, we use a Chinese-English parallel corpus and ask workers to alignment the words in Chinese sentence to the words in English sentence. We do not provide any alignment links from automatic aligner.

2.1 Guidelines of design

MTurk represents a new pattern of market that has yet be thoroughly studied. Mason and Watts (2009) shows that higher payment does not guarantee results with higher quality. Also, one should be aware that the web-based interface is vulnerable to automatic scripts that generate highly consistent yet meaningless results. To ensure a better result, several measures must be combined: 1) Require workers to take qualifications before they can accept the tasks. 2) Implement an interface less vulnerable to automatic scripts. 3) Build quality control mechanism that filters inaccurate results, and finally 4) Redesign the interface so that the time spent by careful and careless workers does not differ too much, so there is less incentives for workers to submit random results. With these guidelines in mind, we put together several elements into the HIT.

Qualifications

We require the workers to take qualifications, which requires them to pick correct translation of five Chinese words. The Chinese word is rendered in bitmap.

Interface implementation

We implemented the word alignment interface on top of Google Web Toolkit, which enables developing `Javascript` based Web application in `Java`. Because all the content of the interface, including the content in the final result, is generated dynamically in the run time, it is much more difficult to hack than plain HTML forms. Figure 1 shows a snapshot of the interface[1]. The labeling procedure requires only mouse click. The worker need to label all the words with a golden background[2]. To complete the task, the worker needs to: 1) Click on the Chinese word he want to label. 2) Click on the English words he want the Chinese word to be linked, or click on the empty word to the end of the sentence. 3) If he want to delete a link, he need to click on the English word again, otherwise he can move on to next unlabeled word, or to modify links on another labeled word. 4) Only when all required words are labeled, the user would be allowed to click on submit button.

The interface has two more functionalities, first, it allows to specify a subset of words in the sentence for user to label, as shown in the snapshot, words with white background are not required to label. Secondly it supports providing initial alignment on the sentence.

Quality control

Quality control is a crucial component of the system. For problems that have clear gold standard answers to a portion of data, the quality control can be done by mingling the known into the unknown, and rejecting the submissions with low qualities on known samples. However in our situation it is not easy to do so because although we have fully manual aligned sentences, we do not have corpus in which the sentences are partially aligned, therefore if we want to use the method we have to let worker label an additional sentence, which may double the effort for the workers. Also we do not provide thorough standard for users, therefore before we know the divergence of the alignments, we actually do not know how to set the threshold, even with given gold standard labels. In addition, if the method will be applied on languages with low resource, we cannot assume availability of gold standard answers. Therefore, we only try to filter out answers base on the consensus. The quality control works as follows. Firstly we assign an alignment task to $2n + 1$ workers. For these submissions, we first try to build a majority answer from these assignments. For each alignment *link*, if it appears in more than n submissions. Then every individual assignments will be compared to the majority alignment, so we can get the precision and recall rates. If either precision or recall rate is lower than a threshold, we will reject the submission.

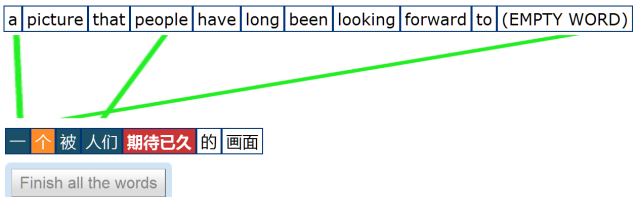

Figure 1: A snapshot of the labeling interface.

2.2 Pricing and worker base

We tried two pricing strategies. The first one fixes the number of words that a worker need to label for each HIT, and fix the rate for each HIT. The second one always

[1] A demo of the latest version can be found at `http://alt-aligner.appspot.com`, the source code of the aligner is distributed under Apache License 2.0 on `http://code.google.com/p/alt-aligner/`

[2] If the document is printed in greyscale, the lightest background (except the white one) is actually golden, the second lightest one is red and the darkest one is dark blue.

asks workers to label every word in the sentence, in the mean time we vary the rate for each HIT according to the lengths of source sentences. For each strategy we tried different rates, starting from 10 words per cent. However we did not get enough workers even after the price raised to 2 words per cent. The result indicates a limited worker base of Chinese speakers.

3 Utilizing the manual alignments

As we can expect, given no explicit guideline for word alignments, the variance of different assignments can be fairly large, a question will raise what can we do with the disagreements? As we will see later in the experiment part, the labels are more likely to be consistent with expert labels if more workers agree on it. Therefore, a simple strategy is to use only the links that more workers have consensus on them.

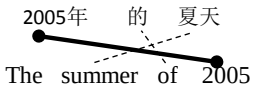

Figure 2: Partial and full alignments

However the method instantly gives rise to a problem. Now the alignment is not "full alignments", instead, they are "partial". The claim seems to be trivial but they have completely different underlying assumptions. Figure 2 shows the comparison of partial alignments (the bold link) and full alignments (the dashed and the bold links). In the example, if full alignment is given, we can assert *2005* is only aligned to 2005年, not to 的or 夏天, but we cannot do that if only partial alignment is given. In this paper we experiment with a novel method which uses the partial alignment to constraint the EM algorithm in the parameter estimation of IBM models.

IBM Models (Brown et. al., 1993) are a series of generative models for word alignment. GIZA++ (Och and Ney, 2003) is the most widely used implementation of IBM models and HMM (Vogel et al., 1996) where EM algorithm is employed to estimate the model parameters. In the E-step, it is possible to obtain sufficient statistics from all possible alignments for simple models such as Model 1 and Model 2. Meanwhile for fertility-based models such as Model 3, 4, 5, enumerating all possible alignments is NP-complete. In practice, we use simpler models such as HMM or Model 2 to generate a "center alignment" and then try to find better alignments among the neighbors of it. The neighbors of an alignment $a_1^J = [a_1, a_2, \cdots, a_J], a_j \in [0, I]$ is defined as alignments that can be generated from a_1^J by one of the operators: 1) Move operator $m_{[i,j]}$, that changes $a_j := i$, i.e. arbitrarily set word f_j in source sentence to align to word e_i in target sentence; 2) Swap operator $s_{[j_1,j_2]}$ that exchanges a_{j_1} and a_{j_2}. The algorithm will update the center alignment as long as a better alignment can be found, and finally outputs a local optimal alignment. The neighbor alignments of the alignment are then used in collecting the counts for the M Step.

In order to use partial manual alignments to constrain the search space, we separate the algorithm into two stages, first the seed alignment will be optimized towards the constraints. Each iteration we only pick a new center alignment with less inconsistent links than the original one, until the alignment is consistent with all constraints. After that, in each iteration we pick the alignment with highest likelihood but does not introduce any inconsistent links. The algorithm will output a local optimal alignment consistent with the partial alignment. When collecting the counts for M-step, we also need to exclude all alignments that are not consistent with the partial manual alignment. The task can also be done by skipping the inconsistent alignments in the neighborhood of the local optimal alignment.

4 Experiment and analysis

In this section we will show the analysis of the harvested MTurk alignments and the results of the semi-supervised word alignment experiments.

4.1 Consistency of the manual alignments

We first examine the internal consistency of the MTurk alignments. We calculate the internal consistency rate in both results. Because we requested three assignments for every question, we classify the links in two different ways. First, if a link appear in all three submissions, we classify it as "consensus link". Second, if a link appear in more than one submissions, we classify it as "majority", otherwise it is classified as "minority". Table 1 presents the statistics of partial alignment and full alignment tasks. Note that by spending the same amount of money, we get more sentences aligned because for fixed rate partial sentence alignment tasks, sometimes we may have overlaps between tasks. Therefore we also calculate a subset of full alignment tasks that consists of all the sentences in partial alignment tasks. The statistics shows that although generally full alignment tasks generates more links, the partial alignment tasks gives denser alignments. It is interesting to know whether the denser alignments lead to higher recall rate or lower precision.

4.2 Comparing MTurk and expert alignments

To exam the quality of alignments, we compared them with expert alignments. Table 2 lists the precision, recall and F-1 scores for partial and full alignment tasks. We compare the consistency of all links, the links in majority group and the consensus links.

As we can observe from the results, the Turkers tend to label less links than the experts, Interestingly, the overall quality of partial alignment tasks is significantly better than full alignment tasks. Despite the lower recall rate, it is encouraging that the majority vote and consensus links

	Partial	Full	Full-Int
Number of sentences	135	239	135
Number of words	2,008	3,241	2,008
Consensus words	13,03	2,299	1,426
Consensus rate(%)	64.89	70.93	71.02
Total Links	7,508	9,767	6,114
Consensus Links	5,625	7,755	4,854
Consensus Rate(%)	74.92	79.40	79.39
Total Unique Links	3,186	3,989	2,506
Consensus Links	1,875	2,585	1,618
Consensus Rate(%)	58.85	64.80	64.54
In majority group	2,447	3,193	1,426
Majority rate(%)	76.80	80.04	71.06

Table 1: Internal consistency of manual alignments, here Full-Int means statistics of full alignment tasks on the sentences that also aligned using partial alignment task

	All Links			Majority Links			Consensus Links		
	P.	R.	F.	P.	R.	F.	P.	R.	F.
P	0.84	0.88	0.86	0.95	0.76	0.84	0.98	0.60	0.74
F	0.88	0.70	0.78	0.96	0.61	0.75	0.99	0.51	0.68
I	0.87	0.71	0.79	0.95	0.62	0.75	0.98	0.52	0.68

Table 2: Consistency of MTurk alignments with expert alignments, showing precision (P), recall (R) and F1 (F) between MTurk and expert alignments. P, F, and I correspond to Partial, Full and Full-Int in Table 1

yield very high precisions against expert alignments. Table 3 lists the words with most errors. Most errors occur on function words. A manual review shows that more than 85% errors have function words on either Chinese side or English side. The result, however, is as expected because these words are hard to label and we did not provide clear rule for function words.

4.3 Results of semi-supervised word alignment

In this experiment we try to use the alignment links in the semi-supervised word alignment algorithm. We use Chinese-English manually aligned corpus in the experiments, which contains 21,863 sentence pairs, 424,683 Chinese words and 524,882 English words. First, we use the parallel corpus to train IBM models without any manual alignments, we run 5 iterations of model 1 and HMM, 3 iterations of model 3 and 6 iterations of model 4. Then we resume the training procedure from the third iterations of model 4. This time we load the manual alignment links and perform 3 iterations of constrained EM. We also experiment with 3 different sets of alignments. Table 4 presents the improvements on the alignment quality.

Chinese				English			
FN		FP		FN		FP	
64	的	16	,	122	the	15	,
26	是	11	个	67	NULL	11	a
19	,	9	是	43	of	6	the
17	个	3	被	36	to	6	is
16	公园	3	有	24	a	4	to

Table 3: Words that most errors occur, FN means a false negative error occurred on the word, i.e. a link to this word or from this word is missing. FP means false positive, accordingly. The manual alignment links comes from majority vote.

Unsupervised						
	Ch-En			En-Ch		
	Prec.	Recall	AER	Prec.	Recall	AER
	68.22	46.88	44.43	65.35	55.05	40.24
All Links						
Partial	68.28	47.09	44.26	65.86	55.63	39.68
Full-Int	68.28	47.09	44.26	65.85	55.63	39.69
Full	68.37	47.15	44.19	65.90	55.67	39.65
Majority Links						
Partial	68.28	47.08	44.27	65.84	55.62	39.70
Full-Int	68.28	47.08	44.27	65.84	55.61	39.71
Full	68.37	47.13	44.20	65.88	55.65	39.67
Consensus Links						
Partial	68.24	47.06	44.30	65.83	55.60	39.71
Full-Int	68.25	47.06	44.29	65.83	55.60	39.72
Full	68.31	47.10	44.25	65.86	55.63	39.68

Table 4: The performance of using manual alignments in semi-supervised word alignment

From the result we can see that given the same amount of links the improvement of alignment error rate is generally the same for partial and full alignment tasks, however, if we consider the amount of money spent on the task, the full alignment task collect much more data than partial alignments, we consider full sentence alignment more cost efficient in this sense.

5 Conclusion

In this pilot experiment, we explore the possibility of using Amazon Mechanical Turk (MTurk) to collect bilingual word alignment data to assist automatic word alignment. We develop a system including a word alignment interface based on Javascript and a quality control scheme. To utilize the manual alignments, we develop a semi-supervised word alignment algorithm that can perform constrained EM with partial alignments. The algorithm enables us to use only the most reliable links by majority vote or consensus. The effectiveness of these methods is proven by small-scale experiments. The results show the manual alignments from MTurk have high precision with expert word alignment, especially when filtered by majority vote or consensus. We get small improvement on semi-supervised word alignment. Given the promising results, it is interesting to see if the tendency will carry on when we scale up the experiments.

However the experiment also shows some problems, first the coverage of worker base on MTurk is limited. Given small worker base for specific languages, the cost efficiency for NLP tasks in those languages is questionable.

References

P. Blunsom and T. Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72.

P. F. Brown et. al. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331.

C. Callison-Burch, D. Talbot, and D. Osborne. 2004. Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004)*.

C. 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*, pages 286–295. Association for Computational Linguistics.

A. Fraser and D. Marcu. 2006. Semi-supervised training for statistical word alignment. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 769–776.

M. Kaisser and J. B. Lowe. 2008. A research collection of question answer sentence pairs. In *Proceedings of The 6th Language Resources and Evaluation Conference*.

M. Kaisser, M. Hearst, and J.B. Lowe. 2008. Evidence for varying search results summary lengths. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

A. Kittur, E. H. Chi, and B Suh. 2008. Crowdsourcing user studies with mechanical turk. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456.

Y. Liu, Q. Liu, and S. Lin. 2005. Log-linear models for word alignment. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 459–466.

A. Lopez and P. Resnik. 2006. Word-based alignment, phrase-based translation: What's the link? with philip resnik. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA-2006)*.

W. Mason and D. J. Watts. 2009. Financial incentives and the "performance of crowds". In *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 77–85.

R. C Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 81–88.

J. Niehues and S. Vogel. 2008. Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 1:29, pages 19–51.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM based word alignment in statistical machine translation. In *Proceedings of 16th International Conference on Computational Linguistics)*, pages 836–841.