Turker-Assisted Paraphrasing for English-Arabic Machine Translation

Michael Denkowski and **Hassan Al-Haj** and **Alon Lavie**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15232, USA
{mdenkows,hhaj,alavie}@cs.cmu.edu

Abstract

This paper describes a semi-automatic paraphrasing task for English-Arabic machine translation conducted using Amazon Mechanical Turk. The method for automatically extracting paraphrases is described, as are several human judgment tasks completed by Turkers. An ideal task type, revised specifically to address feedback from Turkers, is shown to be sophisticated enough to identify and filter problem Turkers while remaining simple enough for non-experts to complete. The results of this task are discussed along with the viability of using this data to combat data sparsity in MT.

1 Introduction

Many language pairs have large amounts of parallel text that can be used to build statistical machine translation (MT) systems. For such language pairs, resources for system tuning and evaluation tend to be disproportionately abundant in the language typically used as the target. For example, the NIST Open Machine Translation Evaluation (OpenMT) 2009 (Garofolo, 2009) constrained Arabic-English development and evaluation data includes four English translations for each Arabic source sentence, as English is the usual target language. However, when considering this data to tune and evaluate an English-to-Arabic system, each English sentence has a single Arabic translation and such translations are often identical. With at most one reference translation for each source sentence, standard minimum error rate training (Och, 2003) to the BLEU metric (Papineni et al., 2002) becomes problematic, as BLEU relies on the availability of multiple references.

We describe a semi-automatic paraphrasing technique that addresses this problem by identifying paraphrases that can be used to create new reference translations based on valid phrase substitutions on existing references. Paraphrases are automatically extracted from a large parallel corpus and filtered by quality judgments collected from human annotators using Amazon Mechanical Turk. As Turkers are not trained to complete natural language processing (NLP) tasks and can dishonestly submit random judgments, we develop a task type that is able to catch problem Turkers while remaining simple enough for untrained annotators to understand.

2 Data Set

The parallel corpus used for paraphrasing consists of all Arabic-English sentence pairs in the NIST OpenMT Evaluation 2009 (Garofolo, 2009) constrained training data. The target corpus to be paraphrased consists of the 728 Arabic sentences from the OpenMT 2002 (Garofolo, 2002) development data.

2.1 Paraphrase Extraction

We conduct word alignment and phrase extraction on the parallel data to produce a phrase table containing Arabic-English phrase pairs (a, e) with translation probabilities $P(a|e)$ and $P(e|a)$. Follow-

ing Bannard and Callison-Burch (2005), we identify Arabic phrases (a_1) in the target corpus that are translated by at least one English phrase (e). We identify paraphrase candidates as alternate Arabic phrases (a_2) that translate e. The probability of a_2 being a paraphrase of a_1 given foreign phrases e is defined:

$$P(a_2|a_1) = \sum_e P(e|a_1)P(a_2|e)$$

A language model trained on the Arabic side of the parallel corpus is used to further score the possible paraphrases. As each original phrase (a_1) occurs in some sentence (s_1) in the target corpus, a paraphrased sentence (s_2) can be created by replacing a_1 with one of its paraphrases (a_2). The final paraphrase score considers context, scaling the paraphrase probability proportionally to the change in log-probability of the sentence:

$$F(a_2, s_2|a_1, s_1) = P(a_2|a_1)\frac{\log P(s_1)}{\log P(s_2)}$$

These scores can be combined for each pair (a_1, a_2) to obtain overall paraphrase scores, however we use the F scores directly as our task considers the sentences in which paraphrases occur.

3 Turker Paraphrase Assessment

To determine which paraphrases to use to transform the development set references, we elicit binary judgments of quality from human annotators. While collecting this data from experts would be expensive and time consuming, Amazon's Mechanical Turk (MTurk) service facilitates the rapid collection of large amounts of inexpensive data from users around the world. As these users are not trained to work on natural language processing tasks, any work posted on MTurk must be designed such that it can be understood and completed successfully by untrained annotators. Further, some Turkers attempt to dishonestly profit from entering random answers, creating a need for tasks to have built-in measures for identifying and filtering out problem Turkers.

Our original evaluation task consists of eliciting two yes/no judgments for each paraphrase and corresponding sentence. Shown the original phrase (a_1) and the paraphrase (a_2), annotators are asked whether or not these two phrases could have the same meaning in some possible context. Annotators are then shown the original sentence (s_1) and the paraphrased sentence (s_2) and asked whether these two sentences have the same meaning. This task has the attractive property that if s_1 and s_2 have the same meaning, a_1 and a_2 *can* have the same meaning. Annotators assigning "yes" to the sentence pair should always assign "yes" to the phrase pair.

To collect these judgments from MTurk, we design a human intelligence task (HIT) that presents Turkers with two instances of the above task along with a text area for optional feedback. The task description asks skilled Arabic speakers to evaluate paraphrases of Arabic text. For each HIT, we pay Turkers \$0.01 and Amazon fees of \$0.005 for a total label cost of \$0.015. For our initial test, we ask Turkers to evaluate the 400 highest-scoring paraphrases, collecting 3 unique judgments for each paraphrase in and out of context. These HITs were completed at a rate of 200 per day.

Examining the results, we notice that most Turkers assign "yes" to the sentence pairs more often than to the phrase pairs, which should not be possible. To determine whether quality of Turkers might be an issue, we run another test for the same 400 paraphrases, this time paying Turkers \$0.02 per HIT and requiring a worker approval rate of 98% to work on this task. These HITs, completed by high quality Turkers at a rate of 100 per day, resulted in similarly impossible data. However, we also received valuable feedback from one of the Turkers.

3.1 Turker Feedback

We received a comment from one Turker that our evaluation task was causing confusion. The Turker would select "no" for some paraphrase in isolation due to missing information. However, the Turker would then select "yes" for the paraphrased sentence, as the context surrounding the phrase rendered the missing information unnecessary. This illustrates the point that untrained annotators understand the idea of "possible context" differently from experts and allows us to restructure our HITs to be ideal for untrained Turkers.

3.2 Revised Main Task

We simplify our task to eliminate as many sources of ambiguity as possible. Our revised task simply presents annotators with the original sentence labeled "sentence 1" and the paraphrased sentence labeled "sentence 2", and asks whether or not the two sentences have the same meaning. Each HIT, titled "Evaluate Arabic Sentences", presents Turkers with 2 such tasks, pays $0.02, and costs $0.005 in Amazon fees.

Without additional consideration, this task remains highly susceptible to random answers from dishonest or unreliable Turkers. To ensure that such Turkers are identified and removed, we intersperse absolute positive and negative examples with the sentence pairs from our data set. Absolute positives consist of the same original sentence s_1 repeated twice and should always receive a "yes" judgment. Absolute negatives consist of some original s_1 and a different, randomly selected original sentence s_1' with several words dropped to obscure meaning. Absolute negatives should always receive a "no" judgment. Positive and negative control cases can be inserted with a frequency based either on desired confidence that enough cases are encountered for normalization or on the availability of funds.

Inserting either a positive or negative control case every 5th task increases the per-label cost to $0.0156. We use this task type to collect 3 unique judgments for each of the 1280 highest-scoring paraphrases at a total cost of $60.00 for 2400 HITs. These HITs were completed substantially faster at a rate of 500-1000 per day. The results of this task are discussed in section 4.

3.3 Editing Task

We conduct an additional experiment to see if Turkers will fix paraphrases judged to be incorrect. The task extends the sentence evaluation task described in the previous section by asking Turkers who select "no" to edit the paraphrase text in the second sentence such that the sentences have the same meaning. While the binary judgment task is used for filtering only, this editing task ensures a usable data point for every HIT completed. As such, fewer total HITs are required and high quality Turkers can be paid more for each HIT. We run 3 sequential tests for this task, offering $0.02, $0.04, and $0.10 per paraphrase approved or edited.

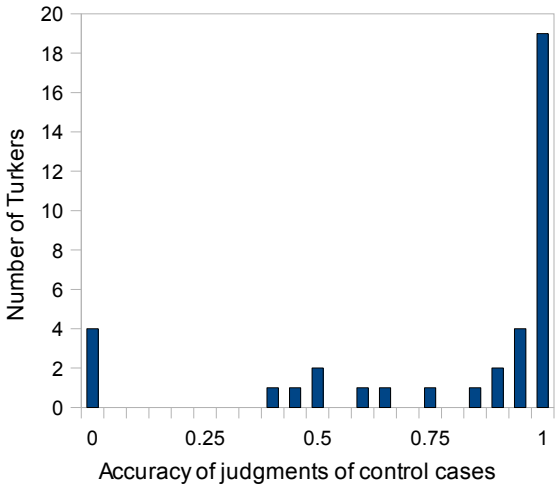

Figure 1: Turker accuracy classifying control cases

Examining the results, we found that regardless of price, very few paraphrases were actually edited, even when Turkers selected "no" for sentence equality. While this allows us to easily identify and remove problem Turkers, it does not solve the issue that honest Turkers either cannot or will not provide usable paraphrase edits for this price range. A brief examination by an expert indicates that the $0.02 per HIT edits are actually better than the $0.10 per HIT edits.

4 Results

Our main task of 2400 HITs was completed through the combined effort of 47 unique Turkers. As shown Figure 1, these Turkers have varying degrees of accuracy classifying the control cases. The two most common classes of Turkers include (1) those spending 15 or more seconds per judgment and scoring above 0.9 accuracy on the control cases and (2) those spending 5-10 seconds per judgment and scoring between 0.4 and 0.6 accuracy as would be expected by chance. As such, we accept but do not consider the judgments of Turkers scoring between 0.7 and 0.9 accuracy on the control set, and reject all HITs for Turkers scoring below 0.7, republishing them to be completed by other workers.

Decision	Confirm	Reject	Undec.
Paraphrases	726	423	131

Table 1: Turker judgments of top 1280 paraphrases

لسلاح	الطيران	الكيني
to-weapon	the-aviation	the-Kenyan
For the Kenyan air force		
للقوات	الجوية	الكينية
for-the-forces	Air	the-Kenyan
For the Kenyan air force		
يلي	اهم	العناوين
following	most-important	headlines
Following are the most important headlines		
يلي	اهم	الانباء
following	most-important	news
Following are the most important headlines		

Figure 2: Paraphrases confirmed by Turkers

After removing judgments from below-threshold annotators, all remaining judgments are used to confirm or reject the covered paraphrases. If a paraphrase has at least 2 remaining judgments, it is confirmed if at least 2 annotators judge it positively and rejected otherwise. Paraphrases with fewer than 2 remaining judgments are considered undecidable. Table 1 shows the distribution of results for the 1280 top-scoring paraphrases. As shown in the table, 726 paraphrases are confirmed as legitimate phrase substitutions on reference translations, providing an average of almost one paraphrase per reference. Figures 2 and 3 show example Arabic paraphrases filtered by Turkers.

5 Conclusions

We have presented a semi-automatic paraphrasing technique for creating additional reference translations. The paraphrase extraction technique provides a ranked list of paraphrases and their contexts which can be incrementally filtered by human judgments. Our judgment task is designed to address specific Turker feedback, remaining simple enough for non-experts while successfully catching problem users. The $60.00 worth of judgments collected produces enough paraphrases to apply an average

النمسا	باسم	
Austria	in-the-name	
Austria on behalf		
من	الرئاسة	النمساوية
from	the-presidency	the-Austrian
From the Austrian presidency		
الدفاع	وزير	
the-defense	minister	
The minister of defense		
الماضي	صرح	وزير	الدفاع
the-past	stated	minister	the-defense
the past the minister of defense stated			

Figure 3: Paraphrases rejected by Turkers

of one phrase substitution to each reference. Our future work includes collecting sufficient data to substitute multiple paraphrases into each Arabic reference in our development set, producing a full additional set of reference translations for use tuning our English-to-Arabic MT system. The resulting individual paraphrases can also be used for other tasks in MT and NLP.

Acknowledgements

This work was supported by a $100 credit from Amazon.com, Inc. as part of a shared task for the NAACL 2010 workshop "Creating Speech and Language Data With Amazon's Mechanical Turk".

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proc. of ACL.*

John Garofolo. 2002. NIST OpenMT Eval. 2002. http://www.itl.nist.gov/iad/mig/tests/mt/2002/.

John Garofolo. 2009. NIST OpenMT Eval. 2009. http://www.itl.nist.gov/iad/mig/tests/mt/2009/.

Franz Josef Och. 2003. Minimum Error Rate Training for Statistical Machine Translation. In *Proc. of ACL.*

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL.*

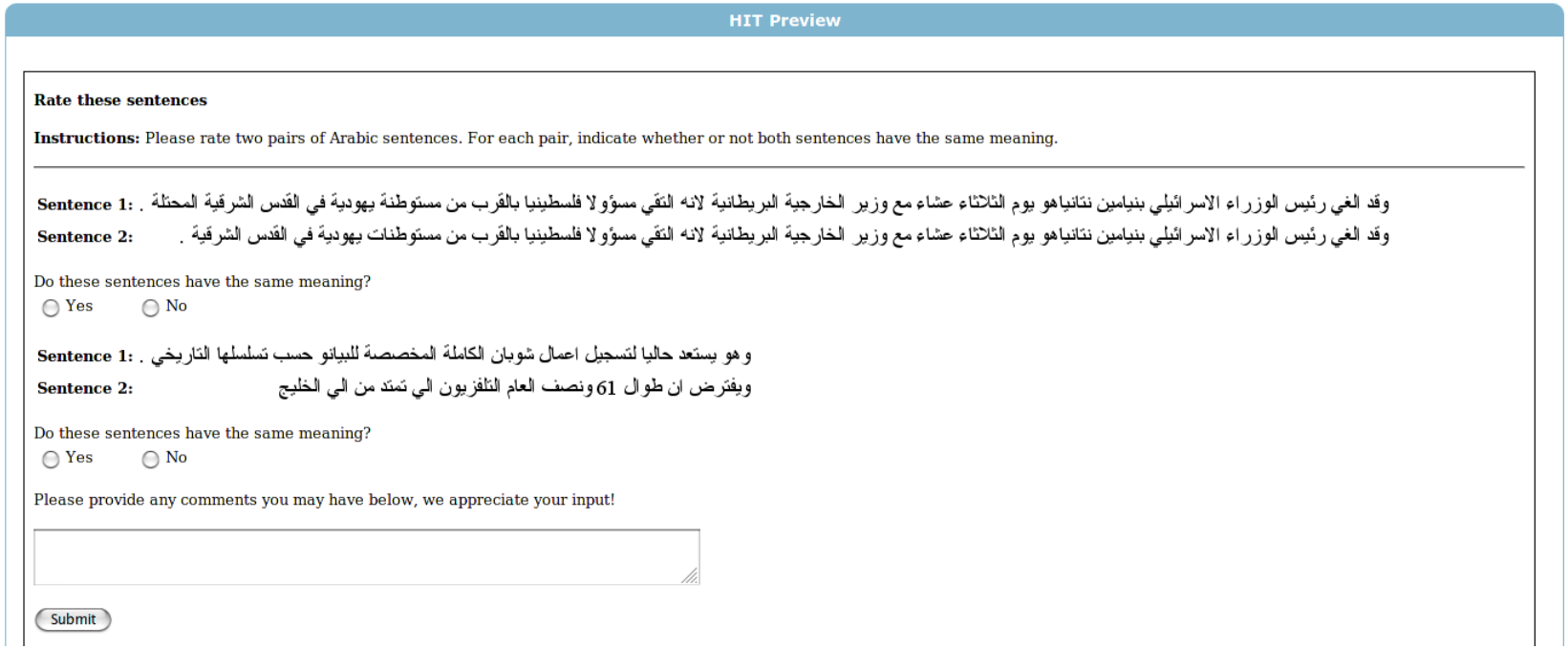

Figure 4: Example HIT as seen by Turkers