Error Driven Paraphrase Annotation using Mechanical Turk

Olivia Buzek
Computer Science and Linguistics
University of Maryland
College Park, MD 20742, USA
olivia.buzek@gmail.com

Philip Resnik
Linguistics and UMIACS
University of Maryland
College Park, MD 20742, USA
resnik@umd.edu

Benjamin B. Bederson
Computer Science and HCIL
University of Maryland
College Park, MD 20742, USA
bederson@cs.umd.edu

Abstract

The source text provided to a machine translation system is typically only one of many ways the input sentence could have been expressed, and alternative forms of expression can often produce a better translation. We introduce here error driven paraphrasing of source sentences: instead of paraphrasing a source sentence exhaustively, we obtain paraphrases for only the parts that are predicted to be problematic for the translation system. We report on an Amazon Mechanical Turk study that explores this idea, and establishes via an oracle evaluation that it holds the potential to substantially improve translation quality.

1 Introduction

The source text provided to a translation system is typically only one of many ways the input sentence could have been expressed, and alternative forms of expression can often produce better translation. This observation is familiar to most statistical MT researchers in the form of preprocessing choices — for example, one segmentation of a Chinese sentence might yield better translations than another.[1] Over the past several years, MT frameworks have been developed that permit *all* the alternatives to be used as input, represented efficiently as a confusion network, lattice, or forest, rather than forcing selection of a single input representation. This has improved performance when applied to phenomena including segmentation, morphological analysis, and more recently source langage word order (Dyer, 2007; Dyer et al., 2008; Dyer and Resnik, to appear).

We have begun to explore the application of the same key idea beyond low-level processing phenomena such as segmentation, instead looking at alternative expressions of meaning. For example, consider translating *The Democratic candidates stepped up their attacks during the debate*. The same basic meaning could have been expressed in many different ways, e.g.:

- During the debate the Democratic candidates stepped up their attacks.
- The Democratic contenders ratcheted up their attacks during the debate.
- The Democratic candidates attacked more aggressively during the debate.
- The candidates in the Democratic debate attacked more vigorously.

These examples illustrate lexical variation, as well as syntactic differences, e.g. whether the attacking or the increasing serves as the main verb. We hypothesize that variation of this kind holds a potential advantage for translation systems, namely that some variations may be more easily translated than others depending on the training data that was given to the system, and we can improve translation quality by allowing a system to take best advantage of the variations it knows about, at the subsentential level, just as the systems described above can take advantage of alternative segmentations.

Paraphrase lattices provide a way to make this hypothesis operational. This idea is a variation on the uses of paraphrase in translation introduced by Callison-Burch and explored by others, as well (Callison-Burch et al., 2006; Madnani et al., 2007; Callison-Burch, 2008; Marton et al., 2009). These authors have shown that performance improvements can be gained by exploiting paraphrases using phrase pivoting. We have investigated using pivoting to create exhaustive paraphrase lattices, and we have also investigated defining upper bounds by eliciting human sub-sentential paraphrases using Mechanical Turk. Unfortunately, in both cases, we have found the size of the paraphrase lattice prohibitive: there are

[1] Chinese is written without spaces, so most MT systems need to segment the input into words as a preprocessing step.

too many spans to paraphrase to make using Turk cost-effective, and automatically generated paraphrase lattices turn out to be too noisy to produce improved translations.

A potential solution to this problem comes from a different line of work we are pursuing, in which translation is viewed as a collaborative process involving people and machines (Bederson et al., 2010). Here, the idea is that in translating from a source to a target language, source- and target-language speakers who are *not bilingual* can collaborate to improve the quality of automatic translation, via an iterative protocol involving translation, back translation, and the use of a very rich user interface. For example, consider the following translation from English to French by an automatic MT system:

- **Source:** Polls indicate Brown, a state senator, and Coakley, Massachusetts' Attorney General, are locked in a virtual tie to fill the late Sen. Ted Kennedy's Senate seat.
- **System:** Les sondages indiquent Brown, un sénateur d'état, et Coakley, Massachusetts' Procureur général, sont enfermés dans une cravate virtuel à remplir le regretté sénateur Ted Kennedy's siège au Sénat.

Someone with only a semester of college French (one of the authors) can look at this automatic translation, and see that the underlined parts are probably wrong. Changing the source sentence to rephrase the underlined pieces (e.g. changing *Massachusetts' Attorney General* to *the Attorney General of Massachusetts*), we obtain a translation that is still imperfect but is more acceptable:

- **System:** Les sondages indiquent que Brown, un sénateur d'état, et Coakley, le procureur général du Massachusetts, sont enfermés dans une cravate virtuel pourvoir le sige au Sénat de Sen. Ted Kennedy, qui est décédé récemment.

One could imagine (and, indeed, we are building) a visual interface that allows a human participant on the target side to communicate back to a source-side collaborator, in effect saying, "These underlined pieces look like they were translated poorly; can you rephrase the relevant parts of your sentence, and perhaps that will lead to a better translation?"[2]

Putting these ideas together — source paraphrase and identification of difficult regions of input for translation — we arrive at the idea of *error driven paraphrasing* of source sentences: instead of paraphrasing to introduce as much variation as possible everywhere in the sentence, we suggest that instead it makes sense to paraphrase only the parts of a source sentence that are problematic for the translation system. In Section 2 we give a first-pass algorithm for error driven paraphrasing, in Section 3 we describe how this was realized using MTurk, and Sections 4 and 5 provide an oracle evaluation, discussion, and conclusions.

[2]Communicating which parts of the sentence are relevant across languages is being done via projection across languages using word alignments; cf. (Hwa et al., 2001).

2 Identifying source spans with errors

In error driven paraphrasing, the key idea is to focus on source spans that are likely to be problematic for translation. Although in principle one could use human feedback from the target side to identify relevant spans, in this paper we begin with an automatic approach, automatically identifying that are likely to be incorrect via a novel algorithm. Briefly, we automatically translate source F to target E, then back-translate to produce F' in the source language. We compare F and F' using TERp (Snover et al., 2009), a form of string-edit distance that identifies various categories of differences between two sentences, and when at least two consecutive non-P (non-paraphrase) edits are found, we flag their smallest containing syntactic constituent.

In more detail, we posit that areas of F' where there were many edits from F will correspond to areas in where the target translation did not match the English very well. Specifically, deletions (D), insertions (I), and shifts (S) are likely to represent errors, while matches (M) and paraphrases (P) probably represent a fairly accurate translation. Furthermore, we assume that while a single D, S, or I edit might be fairly meaningless, a string of at least 2 of those types of edits is likely to represent a substantive problem in the translation.

In order to identify reasonably meaningful paraphrase units based on potential errors, we rely on a source language constituency parser. Using the parse, we find the smallest constituent of the sentence containing all of the tokens in a particular error string. At times, these constituents can be quite large, even the entire sentence. To weed out these cases, we restrict constituent length to no more than 7 tokens.

For example, given

F **The most recent probe to visit Jupiter** was the Pluto-bound New Horizons spacecraft in late February 2007.

E La investigación más reciente fue la visita de Júpiter a Plutón de la envolvente sonda New Horizons a fines de febrero de 2007.

F' The latest research visit Jupiter was the Pluto-bound New Horizons spacecraft in late February 2007.

spans in the the bolded phrase in F would be identified, based on the TERp alignment and smallest containing constituent as shown in Figure 1.

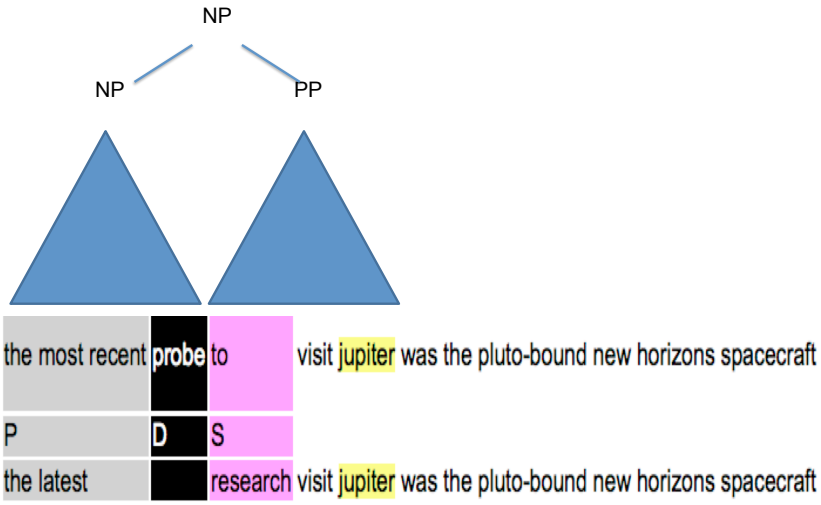

Figure 1: *TERp alignment of a source sentence and its back-translation in order to identify a problematic source span.*

3 Error driven paraphrasing on MTurk

We chose to use translation from English to Chinese in this first foray into Mechanical Turk for error driven paraphrase. This made sense for a number of reasons: first, because we expected to have a much easier time finding Turkers; second, because we could make use of a high quality English parser (in this case the Stanford parser); and, third, because it meant that we as researchers could easily read and judge the quality of Turkers' paraphrases.

To create an English-to-Chinese data set, we used the Chinese-to-English data from the MT08 NIST machine translation evaluation. We used English reference 0 as the source sentence, and the original Chinese sentence as the target. We chose reference 0 because on inspection these references seemed most reflective of native English grammar and usage. The data set comprises 1357 sentence pairs. Using the the above described algorithm to identify possible problem areas in the translation, with the Google Translate API providing both the translation and back-translation, we generated 1780 potential error regions in 1006 of the sentences. Then we created HITs both to obtain paraphrases, and to validate the quality of paraphrase responses. Costs were $117.48 for obtaining multiple paraphrases, and $44.06 for verification.

3.1 Obtaining paraphrases

Based on the phrases marked as problematic by our algorithm, we created HITs asking for paraphrases within 5 sentences, as illustrated in Figure 2. Workers were given 60 minutes to come up with a single paraphrase for each of the five indicated problematic regions, for a reward of $0.10. If a worker felt they could not come up with an alternate phrasing for the marked phrase, they had the option of marking an "Unable to paraphrase" checkbox. We assigned each task to 3 workers, resulting in 3 paraphrases for every marked phrase. From the 1780 errors, we got 5340 responses. Of these, 4821 contained actual paraphrase data, while the rest of the responses indicated an inability to paraphrase, via the checkbox response. All paraphrases were passed on to the verification phase.

3.2 Paraphrase Verification

In the verification phase, we generated alternative full sentences based on the 4821 paraphrases. Workers were shown an original sentence F and asked to compare it to at most 5 alternatives, with a maximum of 20 comparisons made in a HIT. (Recall that although F is the conventional notation for source sentences in machine translation, in this study the F sentences are in English.) Responses were given in the form of radio buttons, marking "Yes" for an alternate sentence if workers felt it was grammatical and accurately reflected the content of the

original sentence, or "No" if it did not meet both of those criteria. Workers were given 30 minutes to make their decisions, for a reward of $0.05. This task was also assigned to 3 workers, resulting in 3 judgments for every paraphrase.

4 Evaluating Results

Using the paraphrase results from Mechanical Turk, we constructed rephrased full sentences for every combination of paraphrase alternatives. For example, if a sentence had 2 sub-spans paraphrased, and the two sub-spans had 2 and 3 unique paraphrasings, respectively, we would construct $2 \times 3 = 6$ alternative full sentences. From the 1780 predicted problematic phrases (within the 1006 automatically identified sentences with possible translation errors), we generated 14,934 rephrased sentences. Each rephrased English sentence was translated into a Chinese sentence, again via the Google Translate API. We then evaluated results for translation of the original sentences, and of all their paraphrase alternatives, via the TER metric, using the MT08 original Chinese sentence as the target-language reference translation. The evaluation set includes the 1000 sentence where at least one paraphrase was provided.[3]

Our evaluation takes the form of an *oracle study*: if we knew with perfect accuracy which variant of a sentence to translate, i.e. among the original and all its paraphrases, based on knowledge of the reference translation, how well could we do? An "oracle" telling us which variant is best is not available in the real world, of course, but in situations like this one, oracle studies are often used to establish the magnitude of the potential gain (Och et al., 2004). In this case, the baseline is the average TER score for the 1000 original sentences, 84.4. If an oracle were permitted to choose which variant was the best to translate, the average TER score would drop to 80.6.[4] Drilling down a bit further, we find that a better-translated paraphrase sentence is available in 313 of the 1000 cases, or31.3%, and for those 313 cases, TER for the best paraphrase alternative improves on the TER for the original sentence by 12.16 TER points.

5 Conclusions

This annotation effort has produced gold standard subsentential paraphrases and paraphrase quality ratings for spans in a large number of sentences, where the choice of spans to paraphrase is specifically focused on regions of the sentence that are difficult to translate. In addition, we have performed an initial analysis, using human-generated paraphrases to provide an oracle evaluation of how much could be gained in translation by translating paraphrases of problematic regions in the source sentence. The results suggest if paraphrasing is automatically targeted to problematic source spans using a back-translation comparison, good paraphrases of the problematic spans could improve translation performance quite substantially.

In future work, we will use a translation system supporting lattice input (Dyer et al., 2008), rather than the Google Translation API, in order to take advantage of fully automatic error-driven paraphrasing, using pivot-based approaches (e.g. (Callison-Burch et al., 2006)) to complete the automation of the error-driven paraphrase process. We will also investigate the use of human rather than machine identification of likely translation problems, in the context of collaborative translation (Bederson et al., 2010).

[3] For the other 6 sentences, all problematic spans were marked "Unable to paraphrase" by all 3 MTurkers.

[4] TER measures errors, so lower is better. A reduction in TER of 3.8 for an MT evaluation dataset would be considered quite substantial; a reduction of 1 point would typically be a publishable result.

References

Benjamin B. Bederson, Chang Hu, and Philip Resnik. 2010. Translation by iterative collaboration between monolingual users. In *Graphics Interface (GI) conference*.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In Robert C. Moore, Jeff A. Bilmes, Jennifer Chu-Carroll, and Mark Sanderson, editors, *HLT-NAACL*. The Association for Computational Linguistics.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *EMNLP*, pages 196–205. ACL.

Chris Dyer and Philip Resnik. to appear. Forest translation. In *NAACL'10*.

C. Dyer, S. Muresan, and P. Resnik. 2008. Generalizing word lattice translation. In *Proceedings of HLT-ACL*, Columbus, OH.

C. Dyer. 2007. Noisier channel translation: translation from morphologically complex languages. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, June.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2001. Evaluating translational correspondence using annotation projection. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 392–399, Morristown, NJ, USA. Association for Computational Linguistics.

Nitin Madnani, Necip Fazil Ayan, Philip Resnik, and Bonnie Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 120–127, Prague, Czech Republic, June. Association for Computational Linguistics.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 381–390, Singapore, August. Association for Computational Linguistics.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir R. Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*, pages 161–168.

Matt Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. TER-Plus: Paraphrases, Semantic, and Alignment Enhancements to Translation Edit Rate. *Machine Translation*.

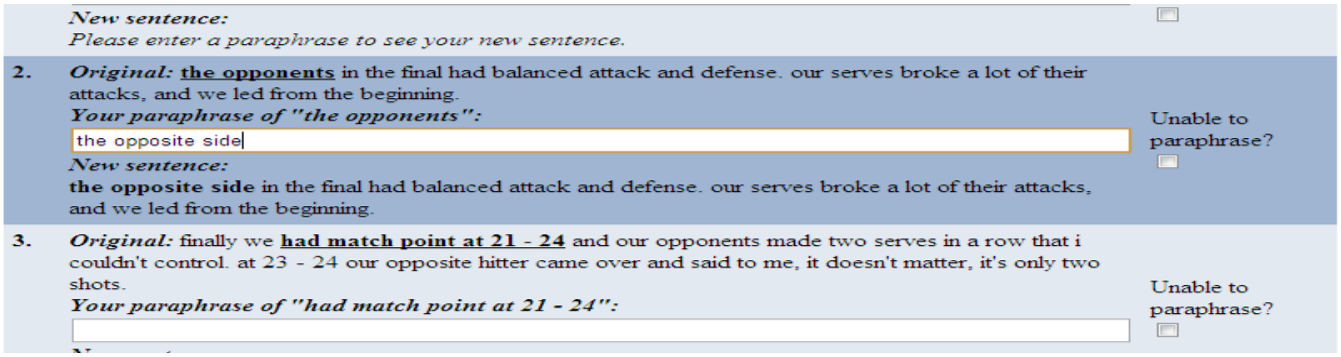

Figure 2: *HIT format 1: Obtaining sub-sentential paraphrases. Note that as the MTurker types a paraphrase into the box,what is typed appears immediately (character by character) in the full-sentence context under "New sentence", so that they can see immediately how the entire sentence looks with their paraphrase.*

the press trust of india quoted

the government minister for relief and rehabilitation kadam

- kadam, the governments relief and rehabilitation minister (2/3)
- the government minister concerned with relief and rehabiliation kadam (1/3)

as revealing today that in the last week, the monsoon has started in

all of indias states one

- every one of indias state, one (3/3)
- each of Indias states one (2/3)
- all states of india one (1/3)

after another, and that the financial losses and casualties have been serious in all areas. just in maharashtra, the state which includes

mumbai, indias largest city,

- india's largest city, mumbai (3/3)
- the largest city in India, Mumbai, (3/3)
- mumbai, the largest city of india, (3/3)

the number of people

known to have died

- who died (3/3)
- identified to have died (2/3)
- known to have passed away (2/3)

has now reached 358.

Figure 3: *Example of error-driven paraphrases produced via HIT format 1, above, for a single sentence. The paraphrase spans (indented) are shown with the number of MTurkers, out of 3, who labeled that paraphrase in context as acceptable using a "validation" HIT.*