Shallow and Deep Paraphrasing for Improved Machine Translation Parameter Optimization

Dennis N. Mehay and Michael White
The Ohio State University
Columbus, Ohio, USA
{mehay,mwhite}@ling.ohio-state.edu

Abstract

String comparison methods such as BLEU (Papineni et al., 2002) are the de facto standard in MT evaluation (MTE) and in MT system parameter tuning (Och, 2003). It is difficult for these metrics to recognize legitimate lexical and grammatical paraphrases, which is important for MT system tuning (Madnani, 2010). We present two methods to address this: a shallow lexical substitution technique and a grammar-driven paraphrasing technique. Grammatically precise paraphrasing is novel in the context of MTE, and demonstrating its usefulness is a key contribution of this paper. We use these techniques to paraphrase a single reference, which, when used for parameter tuning, leads to superior translation performance over baselines that use only human-authored references.

1 Introduction

Because of their speed, simplicity and portability, string comparison methods such as BLEU (Papineni et al., 2002) have become the de facto standard in automatic MTE, as well as in parameter tuning regimes that make heavy use of evaluation, such as MERT (Och, 2003). Although BLEU can be effective for measuring system-level differences among similar systems (Papineni et al., 2002), the surface nature of BLEU's string comparisons makes it difficult to recognize legitimate morphological, syntactic, lexical and paraphrase variation (Callison-Burch et al., 2006), and such recognition is important for MT tuning (Madnani, 2010). One way to address this issue is to to devise extensions to BLEU (Zhou et al., 2006) or to develop new string-based metrics that account for paraphrase and lexical synonymy such as Meteor (Denkowski and Lavie, 2011). Another way is to pad the reference set automatically with paraphrases of the original references (Owczarzak et al., 2006), possibly also doing so in a way that targets each hypothesis under evaluation (Madnani, 2010). The approach described here draws inspiration from both of these tactics, and uses the Meteor metric and a large corpus of n-grams to extend a reference set in a way that is targeted to the output of a baseline system (Section 2). In addition, we generate word-order variants of both the original and the lexically paraphrased references by using a high-precision, grammar-driven parsing and realization system (Section 3). The use of sentence-level paraphrase (Madnani, 2010) — or a rough-and-ready approximations to it (Dyer et al., 2011) — is not new to MTE or parameter tuning. Using deep, grammatically-driven paraphrase, however, is novel in the context of MTE, and demonstrating its usefulness for parameter tuning is a key contribution of this paper.

Targeted lexical substitutions produce reference translations that are more likely **reachable** and **focused** (relevant) w.r.t. a particular translation system being tuned, while grammatical paraphrase helps ensure **correctness**. These are three qualities that Madnani (2010) has argued are important for MT parameter tuning. In a MERT tuning scenario, we find that both paraphrase methods (lexical and grammatical) lead to improved translation results on two held-out validation sets.

2 A Simple Method for Targeted Lexical Paraphrase

Arguably, metrics such as Meteor, which have high correlation to human judgments (Owczarzak, 2008; Denkowski and Lavie, 2011), should be incorporated into system building pipelines. But BLEU is often the metric of evaluation in cross-system comparisons and hence is usually optimized.[1] The technique presented here allows Meteor's lexical knowledge to be injected into the reference set, and therefore into a BLEU-based tuning regime.

Meteor (Denkowski and Lavie, 2011) aligns hypotheses with their references in a greedy multistage process that matches with word forms, then stems, then lexical synonyms, then automatically derived, multiword paraphrases (Bannard and Callison-Burch, 2005). This process is primarily intended to be used directly for evaluation, but it can also be used for other purposes. For example, a targeted paraphrase of one string can be created by substituting into it some of the aligned words from the other string. In this way a reference translation can be modified to target MT system outputs, and thereby extend the reference set in a way that is relevant for retuning that system, as in Figure 1.

Figure 1 depicts a correct paraphrase, but nothing is preventing ungrammatical substitutions such as "...have been **form** election alliances". To address this concern, we follow Chang and Clark (2010) and only permit substitutions that overlap with their context in a way that forms n-grams that were observed in the large corpus of n-grams (or in the original reference). As an example, picking a setting of n=2, this filter would only allow the substitution of '**the parties**' for the original phrase '**these parties**' if the edge bigrams '**all the**' and '**parties have**' occur in either the n-gram corpus or the original references. Note that not filtering with n-grams is similar to Owczarzak et al.'s (2006) method, where paraphrases were mined from the reference set.

Chang and Clark (2010) use the Google n-gram corpus (Brants and Franz, 2006), and they find that a value of n=2 performs best, with an F-measure of 76%. We follow their lead here. They also employ a second, parser-based filter in order to raise the precision of their paraphrase substitutions. We do not use this second filter, as it was not designed to address multiple substitutions in the same sentence.

[1] Och (2003) showed empirically that the metric used for tuning was the one that systems performed best on at test.

3 Precise, Grammatically-Driven, Deep Paraphrasing with OpenCCG

In addition to lexical and multiword paraphrases, string-based MTE metrics also struggle to account for grammatically licensed word-order variation. To enumerate grammatically licensed paraphrases of a reference, we use OpenCCG, an open-source parsing and realization system.[2] OpenCCG features a symbolic-statistical chart parser and surface realizer (White and Rajkumar, 2012). The OpenCCG parser consumes strings and produces semantic dependency graphs (White, 2006), which abstract away from the order of the string. The realizer consumes these graphs and enumerates string realizations that cover all nodes in the graph subject to the grammatical constraints of the CCG syntax. When the parser and realizer are chained end-to-end, OpenCCG becomes a precise, grammatically-driven paraphrase system. As an example, consider the reference *The minister did not however name any associated agency*, which OpenCCG paraphrases as *The minister, however, did not name any associated agency*. This provides a better (uncased) match to one system's output during tuning: *the minister said , however , did not name any assistant organization*. We apply this method both to the original references and to those that have been paraphrased by the method in Section 2.

4 Improved Parameter Optimization in Phrase-Based MT

We use both paraphrase methods described above to test the effects of paraphrasing on MT tuning performance in an Urdu-English translation task. We train phrase-based systems using the Moses toolkit (Koehn et al., 2007). All systems use "hierarchical" lexicalized reordering (Galley and Manning, 2008) and a large distortion limit of 15 to account for the differences in Urdu and English word order. We tune system parameters using MERT with BLEU as the tuning metric. For each experimental condition, we run MERT three times and test for significance

[2] http://openccg.sourceforge.net

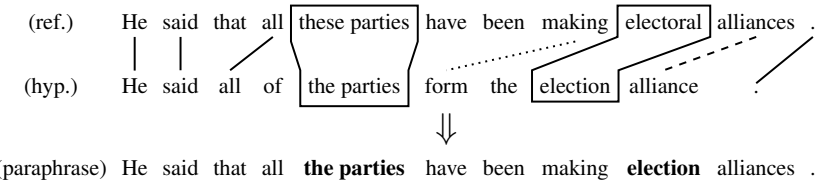

Figure 1: **Top:** Meteor's (v. 1.3) multi-stage alignment using exact matches (solid line), Porter stemming (dashed line), WordNet synonymy (dotted line) and paraphrases (solid outline). **Bottom:** A potential paraphrase.

using Clark et al.'s (2011) method, which helps to control for MERT's inherent instability. Our translation data is the OpenMT Urdu-English training set, which was chosen because its corresponding evaluation reference sets have four references per source segment, allowing a direct comparison with multiple human-authored references. For development tuning and testing, we split the OpenMT-08 evaluation data [LDC2010T21] into two balanced halves of ≈900 segments each. For further test data, we use the whole of the OpenMT-09 evaluation data [LDC2010T23], which has 1,792 segments. Results are computed on the evaluation sets using all four original, human-authored references (i.e., with no paraphrase). All that varies between conditions is the reference sets that are used for tuning.

We tune using the following conditions:[3]

BASELINE-4: 4 human-authored references.

BASELINE-1: 1 "BLEU-best", human-authored reference (see below).

PARASUBS2G: BASELINE-1 + substitutions via the method in Section 2 (with bigram filter).

PARASUBSNOFILT: Like PARASUBS2G, but **without** the bigram filter.

REVR: BASELINE-1 + 2-best reverse realizations of it (distinct from the original) via the method from Section 3.

PARASUBS2GREVR: BASELINE-1 + paraphrase substitutions (with bigram filter) + 2-best reverse realizations thereof.

PARASUBSNOFILTREVR: Like PARASUBS2GREVR, but **without** the bigram filter.

[3]Note that it may be possible to obtain further gains by using additional n-best realizations.

The second, "BLEU-best" condition is obtained by selecting, for each segment in the development tuning set, the single reference that has the highest BLEU score w.r.t. the other references that are distinct from it. This approximates picking the reference that is "best" w.r.t. the other references without rewarding exact duplicates. Table 1 lists tuned system results. As expected, adding lexical paraphrases using the method from Section 2 improves both BLEU and Meteor performance. What was unexpected is that **not** applying the bigram filter leads to higher scores than applying it does. This may be because, in addition to filtering out incorrect lexical substitutions, the bigram filter also blocks *correct* substitutions, when their edge bigrams are not found in the Google n-grams. Also unexpected was that the "BLEU-best" single reference case and the cases where it was paraphrased were superior to the multi-reference condition, in contrast to what Madnani (2010) found in a four reference scenario for Chinese-English translation. This might be due to our method of choosing a single reference, or to a peculiarity of the Urdu-English data set. Nevertheless, the combination of Meteor-driven lexical substitution and OpenCCG parsing and realization achieved high Meteor scores and the highest BLEU scores in all cases. Because BLEU has been found to be sensitive to translation fluency (Owczarzak, 2008), we speculate that the higher BLEU scores may indicate that the grammatical paraphrase tuning method is improving the fluency of the output.

5 Conclusion

We have shown how to extend a set of reference translations using lexical and multiword paraphrase substitution, grammatically licensed reordering, or a combination of the two. All of these techniques lead

	DevTest		OpenMT 2009 Eval	
	Bleu↑	Meteor↑	Bleu↑	Meteor↑
Baseline-4	26.5	28.1	30.9	29.9
Baseline-1	26.6	28.5*	31.1	30.3*
ParaSubs2g	26.6	28.7*	31.1	30.4*
ParaSubsNoFilt	26.8	**28.8***	31.7*	**30.7***
RevR	26.8	28.7*	31.8*	30.5*
ParaSubsNoFiltRevR	26.9*	28.7*	**32.1***	30.5*
ParaSubs2gRevR	**27.4***	**28.8***	32.0*	30.5*

Table 1: Uncased Bleu and Meteor scores on the Urdu-English validation sets. Results significantly better than **Baseline-4** ($p \leq 0.05$) have a '*', and the highest scores are boldfaced. See above for abbreviations.

to improved MT parameter tuning, as compared to using only human-authored translations. In future work, we plan to extend these results to other language pairs and to measure correlations to human judgments.

Acknowledgments

This work was supported in part by the Air Force Research Laboratory under a subcontract to FA8750-09-C-0179.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the ACL*, Ann Arbor, MI, USA.

Thorsten Brants and Alex Franz. 2006. Google Research Web 1T 5-gram Corpus Version 1.1 (LDC2002S28). Linguistic Data Consortium, Philidelphia, PA, USA.

Chris M. Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the EACL*, Trento, Italy.

Ching-Yun Chang and Stephen Clark. 2010. Linguistic steganography using automatically generated paraphrases. In *Proceedings of HLT:ACL*, Los Angeles, CA, USA.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of ACL: HLT*, Portland, OR, USA.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of WMT-11*, Edinburgh, U.K.

Chris Dyer, Kevin Gimpel, Jonathan H. Clark, and Noah A. Smith. 2011. The CMU-ARK German-English Translation System. In *Proceedings of WMT-11*, Edinburgh, U.K.

Michel Galley and Christopher D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of EMNLP-08*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL, Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic.

Nitin Madnani. 2010. *The Circle of Meaning: From Translation to Paraphrasing and Back*. Ph.D. thesis, University of Maryland, College Park, MD, USA.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the ACL*, Sapporo, Japan.

Karolina Owczarzak, Declan Groves, Josef Van Genabith, and Andy Way. 2006. Contextual Bitext-Derived Paraphrases in Automatic MT Evaluation. In *Proceedings of WMT-06*.

Karolina Owczarzak. 2008. *A Novel Dependency-Based Evaluation Metric for Machine Translation*. Ph.D. thesis, Dublin City University, Dublin, Ireland.

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

Michael White and Rajakrishnan Rajkumar. 2012. Minimal dependency length in realization ranking. In *Proceedings of the EMNLP-12/Computational Natural Language Learning*, Jeju Island, Korea.

Michael White. 2006. Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. *Research on Language and Computation*, 4(1):39–75.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating Machine Translation Results with Paraphrase Support. In *Proceedings of EMNLP*, Sydney, Australia.