Automated Paraphrase Lattice Creation for HyTER Machine Translation Evaluation

Marianna Apidianaki[*†], **Guillaume Wisniewski**[†],
Anne Cocos[*] and **Chris Callison-Burch**[*]

[†] LIMSI, CNRS, Univ. Paris Sud, Université Paris-Saclay, 91403 Orsay
[*] Computer and Information Science Department, University of Pennsylvania
{marapi,acocos,ccb}@seas.upenn.edu, guillaume.wisniewski@limsi.fr

Abstract

We propose a variant of a well-known machine translation (MT) evaluation metric, HyTER (Dreyer and Marcu, 2012), which exploits reference translations enriched with meaning equivalent expressions. The original HyTER metric relied on hand-crafted paraphrase networks which restricted its applicability to new data. We test, for the first time, HyTER with automatically built paraphrase lattices. We show that although the metric obtains good results on small and carefully curated data with both manually and automatically selected substitutes, it achieves medium performance on much larger and noisier datasets, demonstrating the limits of the metric for tuning and evaluation of current MT systems.

1 Introduction

Human translators and MT systems can produce multiple plausible translations for input texts. To reward meaning-equivalent but lexically divergent translations, MT evaluation metrics exploit synonyms and paraphrases, or multiple references (Papineni et al., 2002; Doddington, 2002; Denkowski and Lavie, 2010; Lo et al., 2012). The HyTER metric (Dreyer and Marcu, 2012) relies on massive reference networks encoding an exponential number of correct translations for parts of a given sentence, proposed by human annotators. The manually built networks attempt to encode the set of all correct translations for a sentence, and HyTER rewards high quality hypotheses by measuring their minimum edit distance to the set of possible translations.

HyTER spurred a lot of enthusiasm but the need for human annotations heavily reduced its applicability to new data. We propose to use an embedding-based lexical substitution model (Melamud et al., 2015) for building this type of reference networks and test, for the first time, the metric with automatically generated lattices (hereafter HyTERA). We show that HyTERA strongly correlates with HyTER with hand-crafted lattices, and approximates the hTER score (Snover et al., 2006) as measured using post-edits made by human annotators. Furthermore, we generate lattices for standard datasets from a recent WMT Metrics Shared Task and perform the first evaluation of HyTER on large and noisier datasets. The results show that it still remains an interesting solution for MT evaluation, but highlight its limits when used to evaluate recent MT systems that make far less errors of lexical choice than older systems.

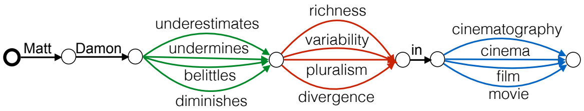

Figure 1: An English reference sentence enriched with substitutes selected by the embedding-based lexical substitution model.

2 The Original HyTER Metric

The HyTER metric (Dreyer and Marcu, 2012) computes the similarity between a translation hypothesis and a reference lattice that compactly encodes millions of meaning-equivalent translations. Formally HyTER is defined as:

$$\text{HyTER}(x, \mathscr{Y}) = \underset{y \in \mathscr{Y}}{\arg\min} \frac{\text{LS}(x, y)}{\text{len}(y)} \tag{1}$$

where $\mathscr{Y}$ is a set of references that can be encoded as a finite state automaton such as the one represented in Figure 1, x is a translation hypothesis and LS is the standard Levenshtein distance, defined as the minimum number of substitutions, deletions and insertions required to transform x into y. We use, in all our experiments, our own im-

Proceedings of NAACL-HLT 2018, pages 480–485
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

plementation of HyTER[1] that relies on the OpenFST framework (Allauzen et al., 2007). Contrary to the original HyTER implementation, we do not consider permutations when transforming x into y as previous results (cf. Table 3 in (Dreyer and Marcu, 2012)) have shown that permutations have only very little impact while significantly increasing the computational complexity of HyTER computation.[2] We also use an exact search rather than a A* search to minimize Equation (1).

The HyTER metric has already been successfully used in MT evaluation but only with handcrafted lattices. To the best of our knowledge, this is the first time it is tested with lattices built automatically.

3 Automatic Lattice Creation

We propose an alternative to the costly manual annotation of reference translations which exploits an embedding-based model of lexical substitution proposed by Melamud et al. (2015) (called AddCos). The original AddCos implementation selects substitutes for words in context from the whole vocabulary. Here, we restrict candidate substitutes to paraphrases of words in the Paraphrase Database (PPDB) XXL package (Ganitkevitch et al., 2013).[3]

AddCos quantifies the fit of substitute word s for target word t in context C by measuring the semantic similarity of the substitute to the target, and the similarity of the substitute to the context:

$$\mathrm{AddCos}(s,t,C) = \frac{\cos(s,t) + \sum_{c\in C}\cos(s,c)}{|C|+1} \quad (2)$$

The vectors s and t are word embeddings of the substitute and target generated by the *skip-gram with negative sampling* model (Mikolov et al., 2013b,a).[4] The context C is the set of context embeddings for words appearing within a fixed-width window of the target t in a sentence (we use a window width of 1). The embeddings c are context embeddings generated by *skip-gram*.[5] In our implementation, we train 300-dimensional word and context embeddings over the 4B words in the Annotated Gigaword (AGiga) corpus (Napoles et al., 2012) using the gensim `word2vec` package (Mikolov et al., 2013b,a; Řehůřek and Sojka, 2010).[6]

Each content word token in a sentence is expanded to include all its possible substitutes selected by AddCos in this specific context, and the lattice can take any path from the expanded start token to the expanded end token. We filter the paraphrase candidates according to: a) their PPDB2.0 score, an *out-of-context* measure of paraphrase confidence which denotes the strength of the relation between the paraphrase and the target word (hereafter, PPDBSc) (Pavlick et al., 2015); b) the substitution score assigned to paraphrases in context by the AddCos model (hereafter, AddCosSc), which shows whether the paraphrase is a good fit for the target word *in a specific context*.[7] Figure 1 shows the four highest ranked paraphrases proposed by AddCos for words in the English reference sentence: *Matt Damon downplays diversity in filmmaking*. The sentences *Matt Damon underestimates richness in cinematography* and *Matt Damon belittles pluralism in cinema.* are included among the 48 references encoded in this lattice.

4 Evaluating HyTER with Automatic Substitutions

We assess the quality of HyTERA to evaluate the quality of MT output both at the sentence and the system level. We first use the setting of Dreyer and Marcu (2012), in Section § 5.1, to compare the score estimated by HyTER and HyTERA to hTER scores. In Section § 5.2, we explore whether HyTERA can reliably predict human translation quality scores from the WMT16 Metrics Shared Task.

[1] The code is available at `https://bitbucket.org/gwisniewski/hytera/`

[2] Note that as permutations of interest can be compactly encoded in a fine-state graph (Kumar and Byrne, 2005), the MOVE operation can be easily considered in our code by applying the substitutions to the permutation lattice rather than to the sentence.

[3] PPDB paraphrases come into packages of different sizes (going from S to XXXL): small packages contain high-precision paraphrases while larger ones have high coverage. All are available from `paraphrase.org`

[4] For the moment, we focus on individual content words. In future work, we plan to also annotate longer text segments in the references with multi-word PPDB paraphrases.

[5] In the original implementation, Melamud et al. (2015) use syntactic dependencies as contexts. We define context as a fixed-width window of words to avoid the need for dependency parsing.

[6] The `word2vec` training parameters we use are a context window of size 3, learning rate *alpha* from 0.025 to 0.0001, minimum word count 100, sampling parameter $1e^{-4}$, 10 negative samples per target word, and 5 training epochs.

[7] Our implementation of AddCos with PPDB substitutes can be found at `https://github.com/acocos/lexsub_addcos`

5 Comparing HyTER and HyTERA

5.1 Open MT NIST Evaluation

To evaluate the performance of HyTER, Dreyer and Marcu (2012) examine whether it can approximate the hTER score (Snover et al., 2006) that measures the number of edits required to change a system output into its post-edition. hTER scores are a good estimate of translation quality and usefulness, but require each translation hypothesis to be corrected by a human annotator. Dreyer and Marcu (2012) show that it can be closely approximated by HyTER scores. In this section, we reproduce their experiments with HyTERA to see whether it is possible to use automatically-built rather than hand-crafted references to approximate hTER scores.

Data Following Dreyer and Marcu (2012), we consider a subset of the 'progress set' used in the 2010 Open MT NIST evaluation.[8] This corpus is made of 102 Arabic and 102 Chinese sentences. Each sentence is automatically translated into English by five MT systems and these translation hypotheses are post-edited by experienced LDC annotators, allowing us to compute hTER scores. The corpus also contains four references and meaning-equivalent annotations which allow for direct comparison to the original HyTER.

Experimental Setting We build meaning-equivalent lattices by applying the lexical substitution method described in Section 3 to each of the four references associated with a sentence, and considering the union of the resulting lattices. We report results for two kinds of lattices: lattices encoding all lexical substitutes available for a word in PPDB (`allPars`) and lattices of substitutes with PPDBSc>2.3 (`allParsFiltered`) and AddCosSc≥0. As expected, the `allPars` lattices are much larger than the manual and the filtered lattices (cf. Table 1). In all our experiments, all corpora are down-cased and tokenized using standard Moses scripts. hTER scores are computed using TERp (Snover et al., 2009).

Sentence Level Evaluation Table 2 reports the correlation between HyTER, HyTERA and hTER at the sentence level. We also include as a baseline the correlation with the sentence-level BLEU score, estimated by the arithmetic mean of 1 to 4-gram precisions.[9]

	ar2en	zh2en
manual	9,454,542	7.8×10^{9}
`allPars`	1.8×10^{27}	8.5×10^{27}
`allParsFiltered`	10,803	3.3×10^{20}

Table 1: Median number of references generated by each method.

In all cases, there is a high correlation between HyTER, HyTERA and hTER, significantly higher than the correlation between BLEU and hTER. This observation shows that replacing the hand-crafted lattices with automatically built ones has only a moderate impact on the HyTER metric quality: automatic lattices result in a small drop of the correlation when evaluating hypotheses translated from Chinese, and slightly improve it for the Arabic to English condition. Overall HyTERA scores are highly correlated with HyTER scores ($\rho = 0.766$ for Arabic and $\rho = 0.756$ for Chinese). More importantly, considering the filtered lattices allows to significantly reduce computation time compared to the `allPars` ones without hurting the quality estimation capacity of the metric.

System Level Evaluation Figure 2 shows how the five MT systems are ranked by the different metrics we consider, when translating from Arabic to English. All metrics rank the systems in the same order, except from HyTER with `allParsFiltered` that only inverts two systems. Note that the tested systems were selected by NIST to cover a variety of system architectures (statistical, rule-based, hybrid) and performances (Dreyer and Marcu, 2012), which makes distinction between them an easy task for all metrics. The benefits of using a metric like HyTER, which focuses on the word level, are much clearer in the sentence-based evaluation (Table 2).

5.2 WMT Metrics Evaluation

In our second set of experiments, we explore the ability of HyTERA to predict direct human judgments at the sentence level using the setting of the WMT16 Metrics Shared Task (Bojar et al., 2016). We measure the correlation between ad-

[8] The corpus is available from LDC under reference `ldc2014t09`.

[9] More precisely: $\text{sBLEU} = \frac{1}{4}\cdot\sum_{i=1}^{4} p_i$ where p_i is the number of i-grams that appears both in the reference and in the hypothesis divided by the number of i-grams in the reference.

	ar → en	zh → en
HyTER	0.667	0.672
HyTERA(`allPars`)	0.673	0.616
HyTERA(`allParsFiltered`)	0.671	0.605
BLEU	-0.563	-0.556

Table 2: Correlation, calculated by Spearman's ρ, between the scores of translation hypotheses estimated by HyTER and hTER.

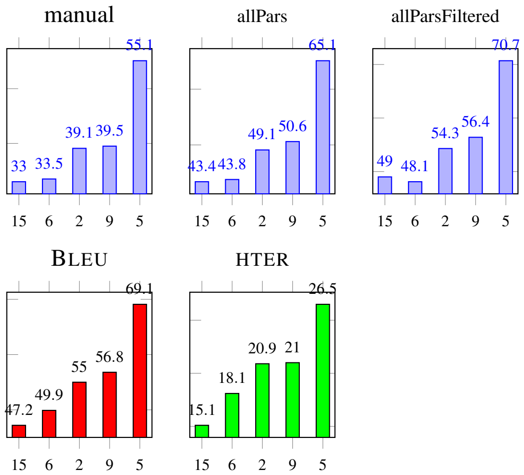

Figure 2: Five Arabic to English MT systems scored by different metrics.

equacy scores collected on Amazon Mechanical Turk following the method advocated by Graham et al. (2016) and the translation quality estimated by applying HyTERA to the official WMT reference. Table 3 reports the results achieved by HyTERA on the six language pairs of the WMT16 Shared Task and its rank among the other metrics tested in the competition.

HyTERA obtains medium performance on the WMT16 dataset, which is much larger and noisier than the dataset used for evaluation in (Dreyer and Marcu, 2012): it is made, for each language, of 560 translations sampled from outputs of all systems taking part in the WMT15 campaign. It is important to note that the hTER scores used in the initial HyTER evaluation were produced by experienced LDC annotators, while the WMT16 Direct Assessment (DA) adequacy judgments were collected from non-experts through crowd-sourcing (Bojar et al., 2016). HyTERA achieves higher performance than the SENTBLEU baseline in four language pairs (cs/de/ru/tr-en). It obtains slightly lower correlation than SENTBLEU for fi-en and ro-en, the language pairs in which correlation was lower for all metrics.

Among the metrics tested at the WMT16 shared task we find combination metrics, and metrics that have been tuned on a development dataset. The metric that performs best for most languages in the segment-level WMT16 evaluation, DPMFCOMB, combines 57 individual metrics (Yu et al., 2015). Similarly, the second highest ranked metric, METRICSF, combines BLEU, METEOR, the alignment-based metric UPF-COMBALT (Fomicheva et al., 2016), and fluency features. The BEER metric, found in fifth position, is a trained evaluation metric with a linear model that combines features capturing character n-grams and permutation trees (Stanojević and Sima'an, 2015).

We report the rank of HyTERA among all metrics (single and combined), and among the single ones. It is important to note that HyTERA needs no tuning, is straightforward to use and very fast to compute, especially with filtered lattices (on average 6s).

The lower performance of the metric on this dataset is also due to the different nature of the MT systems tested. While in the (Dreyer and Marcu, 2012) evaluation, the systems came from the 2010 Open MT NIST evaluation and were selected to cover a variety of architectures and performances, the systems that participated in WMT15 are, for the large part, neural MT systems (Bojar et al., 2015). As reported by Bentivogli et al. (2016), Neural MT systems make at least 17% fewer lexical choice errors than phrase-based systems, which limits the potential of HyTERA, primarily focused on capturing correct lexical choice.

6 Conclusion

We have proposed a method for automatic paraphrase lattice creation which makes the HyTER metric applicable to new datasets. We provide the first evaluation of HyTER on data from a recent

	HyTERA	SENTBleu	WMT All		WMT Single Metrics	
	r	r	best r	rank/15	best r	rank/13
cs → en	.599	.557	.713	10	.671	8
de → en	.498	.448	.601	6	.591	4
fi → en	.476	.484	.598	7	.554	5
ro → en	.483	.499	.662	9	.639	7
ru → en	.525	.502	.618	10	.618	8
tr → en	.540	.532	.663	10	.627	8

Table 3: Pearson correlation between HyTERA and human judgments at the segment level on WMT16 Metrics Shared Task data on different language pairs. We compare to the scores of the SENTBLEU baseline. We report the best correlation achieved by the participating metrics and the rank of HyTERA among all 15 participants, and among the single 13 metrics left after excluding combined ones.

WMT Metrics Shared task. We show that although the metric achieves high correlation with human judgments of translation quality on small and carefully curated data, with both manual and automatically constructed paraphrase networks, it obtains medium performance on recent WMT data. The lower performance is mainly due to the noisier nature of the data and to the higher quality lexical choices made by current neural MT systems, compared to phrase-based and transfer systems, which limits the potential of the metric for system evaluation and tuning.

In its current form, the paraphrase substitution mechanism supports only lexical substitutions. It would be straightforward to extend the AddCos method to handle multi-word paraphrases by training embeddings for multi-word phrases, keeping in mind that longer substitutions might require restructuring the produced sentences to preserve grammaticality.

7 Acknowledgments

We would like to thank Markus Dreyer for sharing with us the original HyTER code.

This work has been supported by the French National Research Agency under project ANR-16-CE33-0013. This material is based in part on research sponsored by DARPA under grant number FA8750-13-2-0017 (the DEFT program) and HR0011-15-C-0115 (the LORELEI program). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA and the U.S. Government.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*. Springer, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, pages 257–267.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal, pages 1–46.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 Metrics Shared Task. In *Proceedings of the First Conference on Machine Translation*. Berlin, Germany, pages 199–231.

Michael Denkowski and Alon Lavie. 2010. Extending the METEOR Machine Translation Evaluation Metric to the Phrase Level. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California, USA, pages 250–253.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*. San Diego, California, pages 138–145.

Markus Dreyer and Daniel Marcu. 2012. HyTER: Meaning-Equivalent Semantics for Translation Evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada, pages 162–171.

Marina Fomicheva, Núria Bel, Lucia Specia, Iria da Cunha, and Anton Malinovskiy. 2016. CobaltF: A Fluent Metric for MT Evaluation. In *Proceedings of the First Conference on Machine Translation*. Berlin, Germany, pages 483–490.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia, pages 758–764.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering* FirstView:1–28.

Shankar Kumar and William Byrne. 2005. Local Phrase Reordering Models for Statistical Machine Translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada, pages 161–168.

Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. 2012. Fully Automatic Semantic MT Evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada, pages 243–252.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A Simple Word Embedding Model for Lexical Substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Denver, Colorado, pages 1–7.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*. Montreal, Canada, pages 95–100.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA, pages 311–318.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China, pages 425–430.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta, pages 45–50.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of AMTA*. Cambridge, MA, USA, pages 223–231.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, Adequacy, or HTER?: Exploring Different Human Judgments with a Tunable MT Metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*. StatMT '09, pages 259–268.

Miloš Stanojević and Khalil Sima'an. 2015. BEER 1.1: ILLC UvA submission to metrics and tuning task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal, pages 396–401.

Hui Yu, Qingsong Ma, Xiaofeng Wu, and Qun Liu. 2015. CASICT-DCU Participation in WMT2015 Metrics Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal, pages 417–421.