Meteor Universal: Language Specific Translation Evaluation for Any Target Language

Michael Denkowski **Alon Lavie**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{mdenkows,alavie}@cs.cmu.edu

Abstract

This paper describes Meteor Universal, released for the 2014 ACL Workshop on Statistical Machine Translation. Meteor Universal brings language specific evaluation to previously unsupported target languages by (1) automatically extracting linguistic resources (paraphrase tables and function word lists) from the bitext used to train MT systems and (2) using a universal parameter set learned from pooling human judgments of translation quality from several language directions. Meteor Universal is shown to significantly outperform baseline BLEU on two new languages, Russian (WMT13) and Hindi (WMT14).

1 Introduction

Recent WMT evaluations have seen a variety of metrics employ language specific resources to replicate human translation rankings far better than simple baselines (Callison-Burch et al., 2011; Callison-Burch et al., 2012; Macháček and Bojar, 2013; Snover et al., 2009; Denkowski and Lavie, 2011; Dahlmeier et al., 2011; Chen et al., 2012; Wang and Manning, 2012, *inter alia*). While the wealth of linguistic resources for the WMT languages allows the development of sophisticated metrics, most of the world's 7,000+ languages lack the prerequisites for building advanced metrics. Researchers working on low resource languages are usually limited to baseline BLEU (Papineni et al., 2002) for evaluating translation quality.

Meteor Universal brings language specific evaluation to any target language by combining linguistic resources automatically learned from MT system training data with a universal metric parameter set that generalizes across languages. Given only the bitext used to build a standard phrase-based translation system, Meteor Universal learns a paraphrase table and function word list, two of the most consistently beneficial language specific resources employed in versions of Meteor. Whereas previous versions of Meteor require human ranking judgments in the target language to learn parameters, Meteor Universal uses a single parameter set learned from pooling judgments from several languages. This universal parameter set captures general preferences shown by human evaluators across languages. We show this approach to significantly outperform baseline BLEU for two new languages, Russian and Hindi. The following sections review Meteor's scoring function (§2), describe the automatic extraction of language specific resources (§3), discuss training of the universal parameter set (§4), report experimental results (§5), describe released software (§6), and conclude (§7).

2 Meteor Scoring

Meteor evaluates translation hypotheses by aligning them to reference translations and calculating sentence-level similarity scores. For a hypothesis-reference pair, the space of possible alignments is constructed by exhaustively identifying all possible matches between the sentences according to the following matchers:

Exact: Match words if their surface forms are identical.

Stem: Stem words using a language appropriate Snowball Stemmer (Porter, 2001) and match if the stems are identical.

Synonym: Match words if they share membership in any synonym set according to the WordNet database (Miller and Fellbaum, 2007).

Paraphrase: Match phrases if they are listed as

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 376–380,
Baltimore, Maryland USA, June 26–27, 2014.

paraphrases in a language appropriate paraphrase table (described in §3.2).

All matches are generalized to phrase matches with a span in each sentence. Any word occurring within the span is considered covered by the match. The final alignment is then resolved as the largest subset of all matches meeting the following criteria in order of importance:

1. Require each word in each sentence to be covered by zero or one matches.
2. Maximize the number of covered words across both sentences.
3. Minimize the number of *chunks*, where a *chunk* is defined as a series of matches that is contiguous and identically ordered in both sentences.
4. Minimize the sum of absolute distances between match start indices in the two sentences. (Break ties by preferring to align phrases that occur at similar positions in both sentences.)

Alignment resolution is conducted as a beam search using a heuristic based on the above criteria.

The Meteor score for an aligned sentence pair is calculated as follows. Content and function words are identified in the hypothesis (h_c, h_f) and reference (r_c, r_f) according to a function word list (described in §3.1). For each of the matchers (m_i), count the number of content and function words covered by matches of this type in the hypothesis ($m_i(h_c)$, $m_i(h_f)$) and reference ($m_i(r_c)$, $m_i(r_f)$). Calculate weighted precision and recall using matcher weights ($w_i ... w_n$) and content-function word weight (δ):

$$P = \frac{\sum_i w_i \cdot (\delta \cdot m_i(h_c) + (1-\delta) \cdot m_i(h_f))}{\delta \cdot |h_c| + (1-\delta) \cdot |h_f|}$$

$$R = \frac{\sum_i w_i \cdot (\delta \cdot m_i(r_c) + (1-\delta) \cdot m_i(r_f))}{\delta \cdot |r_c| + (1-\delta) \cdot |r_f|}$$

The parameterized harmonic mean of P and R (van Rijsbergen, 1979) is then calculated:

$$F_{mean} = \frac{P \cdot R}{\alpha \cdot P + (1-\alpha) \cdot R}$$

To account for gaps and differences in word order, a fragmentation penalty is calculated using the total number of matched words (m, averaged over hypothesis and reference) and number of chunks (ch):

$$Pen = \gamma \cdot \left(\frac{ch}{m}\right)^{\beta}$$

The Meteor score is then calculated:

$$Score = (1 - Pen) \cdot F_{mean}$$

The parameters α, β, γ, δ, and $w_i ... w_n$ are tuned to maximize correlation with human judgments.

3 Language Specific Resources

Meteor uses language specific resources to dramatically improve evaluation accuracy. While some resources such as WordNet and the Snowball stemmers are limited to one or a few languages, other resources can be learned from data for any language. Meteor Universal uses the same bitext used to build statistical translation systems to learn function words and paraphrases. Used in conjunction with the universal parameter set, these resources bring language specific evaluation to new target languages.

3.1 Function Word Lists

The function word list is used to discriminate between content and function words in the target language. Meteor Universal counts words in the target side of the training bitext and considers any word with relative frequency above 10^{-3} to be a function word. This list is used only during the scoring stage of evaluation, where the tunable δ parameter controls the relative weight of content versus function words. When tuned to match human judgments, this parameter usually reflects a greater importance for content words.

3.2 Paraphrase Tables

Paraphrase tables allow many-to-many matches that can encapsulate any local language phenomena, including morphology, synonymy, and true paraphrasing. Identifying these matches allows far more sophisticated evaluation than is possible with simple surface form matches. In Meteor Universal, paraphrases act as the catch-all for non-exact matches. Paraphrases are automatically extracted from the training bitext using the translation pivot approach (Bannard and Callison-Burch, 2005). First, a standard phrase table is learned from the bitext (Koehn et al., 2003). Paraphrase extraction then proceeds as follows. For each target language phrase (e_1) in the table, find each

source phrase f that e_1 translates. Each alternate phrase ($e_2 \neq e_1$) that translates f is considered a paraphrase with probability $P(f|e_1) \cdot P(e_2|f)$. The total probability of e_2 being a paraphrase of e_1 is the sum over all possible pivot phrases f:

$$P(e_2|e_1) = \sum_f P(f|e_1) \cdot P(e_2|f)$$

To improve paraphrase precision, we apply several language independent pruning techniques. The following are applied to each paraphrase instance (e_1, f, e_2):

- Discard instances with very low probability ($P(f|e_1) \cdot P(e_2|f) < 0.001$).
- Discard instances where e_1, f, or e_2 contain punctuation characters.
- Discard instances where e_1, f, or e_2 contain only function words (relative frequency above 10^{-3} in the bitext).

The following are applied to each final paraphrase (e_1, e_2) after summing over all instances:

- Discard paraphrases with very low probability ($P(e_2|e_1) < 0.01$).
- Discard paraphrases where e_2 is a sub-phrase of e_1.

This constitutes the full Meteor paraphrasing pipeline that has been used to build tables for fully supported languages (Denkowski and Lavie, 2011). Paraphrases for new languages have the added advantage of being extracted from the same bitext that MT systems use for phrase extraction, resulting in ideal paraphrase coverage for evaluated systems.

4 Universal Parameter Set

Traditionally, building a version of Meteor for a new target language has required a set of human-scored machine translations, most frequently in the form of WMT rankings. The general lack of availability of these judgments has severely limited the number of languages for which Meteor versions could be trained. Meteor Universal addresses this problem with the introduction of a "universal" parameter set that captures general human preferences that apply to all languages for which judgment data does exist. We learn this parameter set by pooling over 100,000 binary ranking judgments from WMT12 (Callison-Burch et al., 2012) that cover 8 language directions (details in Table 1). Data for each language is scored using the same resources (function word list and paraphrase table only) and scoring parameters are tuned to maximize agreement (Kendall's τ) over all judgments from all languages, leading to a single parameter set. The universal parameter set encodes the following general human preferences:

Direction	Judgments
cs-en	11,021
de-en	11,934
es-en	9,796
fr-en	11,594
en-cs	18,805
en-de	14,553
en-es	11,834
en-fr	11,562
Total	101,099

Table 1: Binary ranking judgments per language direction used to learn parameters for Meteor Universal

- Prefer recall over precision.
- Prefer word choice over word order.
- Prefer correct translations of content words over function words.
- Prefer exact matches over paraphrase matches, while still giving significant credit to paraphrases.

Table 2 compares the universal parameters to those learned for specific languages in previous versions of Meteor. Notably, the universal parameter set is more balanced, showing a normalizing effect from generalizing across several language directions.

5 Experiments

We evaluate the Universal version of Meteor against full language dedicated versions of Meteor and baseline BLEU on the WMT13 rankings. Results for English, Czech, German, Spanish, and French are biased in favor of Meteor Universal since rankings for these target languages are included in the training data while Russian constitutes a true held out test. We also report the results of the WMT14 Hindi evaluation task. Shown

Language	α	β	γ	δ	w_{exact}	w_{stem}	w_{syn}	w_{par}
English	0.85	0.20	0.60	0.75	1.00	0.60	0.80	0.60
Czech	0.95	0.20	0.60	0.80	1.00	–	–	0.40
German	0.95	1.00	0.55	0.55	1.00	0.80	–	0.20
Spanish	0.65	1.30	0.50	0.80	1.00	0.80	–	0.60
French	0.90	1.40	0.60	0.65	1.00	0.20	–	0.40
Universal	0.70	1.40	0.30	0.70	1.00	–	–	0.60

Table 2: Comparison of parameters for language specific and universal versions of Meteor.

WMT13 τ	M-Full	M-Universal	BLEU
English	0.214	0.206	0.124
Czech	0.092	0.085	0.044
German	0.163	0.157	0.097
Spanish	0.106	0.101	0.068
French	0.150	0.137	0.099
Russian	–	0.128	0.068

WMT14 τ	M-Full	M-Universal	BLEU
Hindi	–	0.264	0.227

Table 3: Sentence-level correlation with human rankings (Kendall's τ) for Meteor (language specific versions), Meteor Universal, and BLEU

in Table 3, Meteor Universal significantly outperforms baseline BLEU in all cases while suffering only slight degradation compared to versions of Meteor tuned for individual languages. For Russian, correlation is nearly double that of BLEU. This provides substantial evidence that Meteor Universal will further generalize, bringing improved evaluation accuracy to new target languages currently limited to baseline language independent metrics.

For the WMT14 evaluation, we use the traditional language specific versions of Meteor for all language directions except Hindi. This includes Russian, for which additional language specific resources (a Snowball word stemmer) help significantly. For Hindi, we use the release version of Meteor Universal to extract linguistic resources from the constrained training bitext provided for the shared translation task. These resources are used with the universal parameter set to score all system outputs for the English–Hindi direction.

6 Software

Meteor Universal is included in Meteor version 1.5 which is publicly released for WMT14. Meteor 1.5 can be downloaded from the official webpage[1] and a full tutorial for Meteor Universal is available online.[2] Building a version of Meteor for a new language requires a training bitext (*corpus.f*, *corpus.e*) and a standard Moses format phrase table (*phrase-table.gz*) (Koehn et al., 2007). To extract linguistic resources for Meteor, run the new language script:

```
$ python scripts/new_language.py out \
corpus.f corpus.e phrase-table.gz
```

To use the resulting files to score translations with Meteor, use the new language option:

```
$ java -jar meteor-*.jar test ref -new \
out/meteor-files
```

Meteor 1.5, including Meteor Universal, is free software released under the terms of the GNU Lesser General Public License.

7 Conclusion

This paper describes Meteor Universal, a version of the Meteor metric that brings language specific evaluation to any target language using the same resources used to build statistical translation systems. Held out tests show Meteor Universal to significantly outperform baseline BLEU on WMT13 Russian and WMT14 Hindi. Meteor version 1.5 is freely available open source software.

Acknowledgements

This work is supported in part by the National Science Foundation under grant IIS-0915327, by the Qatar National Research Fund (a member of the Qatar Foundation) under grant NPRP 09-1140-1-177, and by the NSF-sponsored XSEDE program under grant TG-CCR110017.

[1] http://www.cs.cmu.edu/~alavie/METEOR/

[2] http://www.cs.cmu.edu/~mdenkows/meteor-universal.html

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

Boxing Chen, Roland Kuhn, and George Foster. 2012. Improving amber, an mt evaluation metric. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 59–63, Montréal, Canada, June. Association for Computational Linguistics.

Daniel Dahlmeier, Chang Liu, and Hwee Tou Ng. 2011. Tesla at wmt 2011: Translation evaluation and tunable metric. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 78–84, Edinburgh, Scotland, July. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL/HLT 2003*.

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 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 metrics shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria, August. Association for Computational Linguistics.

George Miller and Christiane Fellbaum. 2007. WordNet. http://wordnet.princeton.edu/.

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

Martin Porter. 2001. Snowball: A language for stemming algorithms. http://snowball.tartarus.org/texts/.

Matthew Snover, Nitin Madnani, Bonnie 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*, pages 259–268, Athens, Greece, March. Association for Computational Linguistics.

C. J. van Rijsbergen, 1979. *Information Retrieval*, chapter 7. Butterworths, London, UK, 2nd edition.

Mengqiu Wang and Christopher Manning. 2012. Spede: Probabilistic edit distance metrics for mt evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 76–83, Montréal, Canada, June. Association for Computational Linguistics.