

PyMEANT: Automatic Semantic MT Evaluation in Python

Roger Que

query@jhu.edu

<https://github.com/query/mt-submissions/tree/master/project>

Abstract

In this paper, I present PyMEANT, a Python implementation of the MEANT machine translation evaluation metric (Lo et al., 2012). MEANT matches semantic frames and arguments, as identified by a shallow semantic parse, in order to ensure the preservation of each sentence’s argument structure. The semantic similarity of word pairs is judged by the Jaccard index of their respective sets of co-occurring contexts. Tests show that PyMEANT achieves better pairwise ranking accuracy against human adequacy judgments than BLEU.

1 Introduction

It is generally accepted that human judgments of translation adequacy are ultimately based on the semantic content of a hypothesis. Therefore, an automatic evaluation process that successfully incorporates semantic information should, in theory, obtain better correlation with human judgments than a semantically-ignorant evaluator run on the same data.

Currently, most widely used translation quality metrics are based primarily on measures of n -gram precision and recall, including BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005). This approach has the advantage of being easy to incorporate into translation systems due to its relatively inexpensive scoring method, and does a good job of capturing translation fluency. However, it also fails to sufficiently capture important aspects of a sentence’s meaning. With BLEU, in particular, a hypothesis’ phrases may be permuted around bigram mismatch

sites to yield a second hypothesis that is clearly incoherent, yet still has the same score (Callison-Burch et al., 2006). This presents a problem for evaluation not only on fluency, but on adequacy as well. The implication is that, in many cases, a sentence’s arguments may be reordered in a way that preserves its BLEU score, but completely changes its meaning. It is clear that n -gram measurements alone are not sufficient to ensure that important semantic relations are kept intact, and that more direct inspection of a hypothesis’ semantic content is needed.

However, semantic evaluation is a difficult problem for a number of reasons. Foremost among these issues is the question of how to capture semantic information in a usefully comparable way, so that translation hypotheses may be checked against their respective references. Although proposals for the representation of deep semantic structure, such as AMR’s acyclic graphical notation (Banarescu et al., 2013), and corresponding parsers into these representations (Flanigan et al., 2014) exist, the best method for measuring the similarity of multiple instances of these complex structures is still an open question. Conceptually straightforward methods, such as graph edit distance, are NP-hard and thus present computational tractability problems. Ideally, we would like a metric with relatively low overhead that avoids constructing large graphs or other complex structures to represent each sentence.

2 The MEANT metric

In order to avoid these technical challenges, MEANT takes a simplified approach to the incorporation of semantic information. Instead of attempting to discern the full semantic structure of a sentence, it performs

a shallow semantic parse that identifies a series of frames containing verbs, also referred to as *predicates*, and their semantic role arguments. It then aligns frames with similar predicates together, and computes the total similarity of the predicates and arguments in order to yield a final score.

2.1 Semantic frame identification

The first task in MEANT is the extraction of frames from raw sentences, which is equivalent to a semantic role labeling task. There are several SRL tagging schemata in wide use, chief among them FrameNet (Fillmore et al., 2003) and PropBank (Palmer et al., 2005). These two systems differ primarily in their approach to generalizing over semantic roles from verb to verb.

FrameNet attempts to define generic roles, such as **Addressee** and **Seller**, than can be applied to multiple verbs. This introduces the possibility of weighting arguments based on their role class. For example, with sentences that describe a commercial transaction, we may choose to emphasize the parties involved over the goods exchanged by weighting the former more heavily. However, this also makes the construction of a corpus more difficult, as argument classes must remain consistent over all verbs to which they apply.

PropBank, on the other hand, defines arguments on a verb-by-verb basis, and only informally correlates them across different verbs. For this reason, although **Arg0** is usually a prototypical agent, and **Arg1** is often a patient or theme, it is improper to construe instances of these arguments attached to specific verbs as part of a broader argument class.

PyMEANT uses the PropBank tagging scheme, following the lead of the original MEANT implementation. The ASSERT automatic semantic role tagger (Pradhan et al., 2004) is used to identify individual frames. The predicates of frames identified by ASSERT for a sample translation hypothesis and its corresponding reference are shown in Figure 1.

2.2 Lexical similarity measures

Having identified the relevant predicates and arguments, we would now like to determine how similar they are. Ideally, the measure we use should be able to capture some intuitions about the environments in which semantically related words occur. As men-

MTO: When the reporter went¹ to the hotel at 4:00 p.m., however, most of the job-seekers to participate² in the recruitment, not many people will come³ back, but has been can tell⁴ all days of the South China Sea north of accent.

REF: It was four in the afternoon when our reporter arrived^A at the hostel, but most of the job-seekers were still elsewhere taking^B part in job fairs and not many had returned^C yet. However, one could already hear^E accents from all over the country.

Figure 1: Semantic frame predicates identified by ASSERT for sample machine translation output (MTO) and corresponding reference (REF) from the DARPA GALE corpus.

tioned earlier, n -gram metrics such as BLEU and METEOR are unlikely to be helpful here, particularly because of the short length of the word strings involved.

MEANT relies on the notion that words with similar meaning are likely to appear surrounded by similar contexts, and thus the set of words that they co-occur with should strongly overlap. There are several information-theoretic metrics for this type of *lexical similarity*. PyMEANT uses the Jaccard similarity, which has been shown to be an effective measure for use in MEANT (Tumuluru et al., 2012), and is fairly straightforward to implement. Generally defined, for two sets A and B , the Jaccard similarity is the cardinality of their intersection divided by the cardinality of their union:

$$J(x, y) = \frac{|A \cap B|}{|A \cup B|}$$

For our purposes, A and B are multisets containing the word types with which some word types a and b have co-occurred. We use a co-occurrence window size to restrict the number of possible pairs. PyMEANT’s default window size is 3, meaning that two words must occur together within a single 3-

word span to be considered co-occurring; equivalently, they may be separated by at most one other word. Sizes up to 13 have been tested, though these longer windows yield no corresponding increase in MEANT’s overall accuracy (Tumurluru et al., 2012). Defined, then, in terms of the number of co-occurrences $c(a, b)$ for the words a and b , and the set of all word types W , the Jaccard similarity coefficient $J(a, b)$ is:

$$J(x, y) = \frac{\sum_{w \in W} \min(c(a, w), c(b, w))}{\sum_{w \in W} \max(c(a, w), c(b, w))}$$

2.3 Predicate and argument matching

We now turn to computation of the overall MEANT score. This is accomplished by aligning semantic frames between the hypothesis and the reference, then the words in each of the frames’ arguments, and summing over the lexical similarity of each pair of aligned words.

MEANT aligns individual words with each other by performing maximum bipartite matching over a graph with each node representing a single word. Edges connect each hypothesis node with each reference node, weighted by the lexical similarity of the two words at each end. After matching, each word is aligned to exactly zero or one words on the opposite side of the graph. This process is first used to align frames with each other, using their predicates as the graph nodes. Then, for each pair of frames, the words of arguments with the same label are also aligned to each other, using the words themselves as nodes. A sample predicate alignment graph is illustrated in Figure 2, with the weights used for matching shown in Table 1.

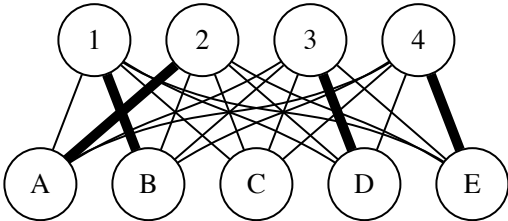


Figure 2: A bipartite graph representation of the possible alignments of the predicates shown in Figure 1. Thick lines indicate the alignments selected by the bipartite matching algorithm.

		MT output			
		1	2	3	4
Reference	A	0.117	0.130	0.090	0.121
	B	0.194	0.085	0.166	0.165
	C	0.165	0.119	0.122	0.194
	D	0.208	0.067	0.238	0.196
	E	0.146	0.127	0.142	0.301

Table 1: Pairwise Jaccard similarity scores for the predicates shown in Figure 1. The alignments selected by the bipartite matching algorithm are highlighted in bold.

The similarity score for each predicate and argument is computed by summing over the Jaccard coefficient for each pair of aligned words, then dividing by word length of either the hypothesis or the reference side, whichever is longer. In algebraic terms, for some word vector \vec{h} representing a predicate or argument in the hypothesis, the corresponding reference-side vector \vec{r} , and a set of alignments $A(\vec{h}, \vec{r})$ containing ordered alignment word pairs (i, j) , the similarity score $S(\vec{h}, \vec{r})$ is defined as:

$$S(\vec{h}, \vec{r}) = \frac{\sum_{(w, x) \in A(\vec{h}, \vec{r})} J(w, x)}{\max(|\vec{h}|, |\vec{r}|)}$$

We now compute a total score for the entire sentence. The score for each frame is weighted by the fraction of words it covers in the overall sentence, such that larger, and thus more semantically important, frames have more influence over the final score than smaller ones. A frame \mathbf{f} ’s coverage ratio in a sentence V , denoted $v(\mathbf{f}, V)$, is equal to:

$$v(\mathbf{f}, V) = \frac{|\mathbf{f}|}{|V|}$$

For each hypothesis H and reference R , separate precision and recall scores $p(H, R)$ and $r(H, R)$ are first computed. The precision is normalized against the maximum possible score given total coverage of the frames in the MT output, while the recall is normalized against the same for the reference. Let $n(\mathbf{f})$ denote the number of arguments to frame \mathbf{f} , including the predicate; let $A(H, R)$ denote the set of frame alignments (\mathbf{f}, \mathbf{g}) between the sentences H and R ; and let $A(\mathbf{f}, \mathbf{g})$ denote the set of argument alignments (\vec{h}, \vec{r}) between frames \mathbf{f} and \mathbf{g} . Then the precision and

recall are:

$$p(H, R) = \sum_{(\mathbf{f}, \mathbf{g}) \in A(H, R)} \left[\frac{v(\mathbf{f}, V)}{n(\mathbf{f})} \cdot \sum_{(\vec{h}, \vec{r}) \in A(\mathbf{f}, \mathbf{g})} S(\vec{h}, \vec{r}) \right]$$

$$r(H, R) = \sum_{(\mathbf{f}, \mathbf{g}) \in A(H, R)} \left[\frac{v(\mathbf{g}, V)}{n(\mathbf{g})} \cdot \sum_{(\vec{h}, \vec{r}) \in A(\mathbf{f}, \mathbf{g})} S(\vec{h}, \vec{r}) \right]$$

The final MEANT score is the F_1 measure:

$$\text{MEANT}(H, R) = 2 \cdot \frac{p(H, R) \cdot r(H, R)}{p(H, R) + r(H, R)}$$

Later revisions of MEANT add tunable parameters that control the relative importance of predicates and argument classes (Lo and Wu, 2013). PyMEANT does not implement this feature, and so all predicates and arguments are weighted equally in the final computation.

3 Results

PyMEANT was tested by measuring the total accuracy of pairwise judgments on the DARPA GALE 2.5 broadcast news (BN) data sets for the Arabic–English and Chinese–English language pairs. For each translation segment, “correct” pairwise rankings between MT systems were obtained by taking all pairs of systems and comparing the mean human adequacy ratings. The PyMEANT lexical similarity model was trained on the April 1995 section of Gigaword’s *New York Times* corpus; due to the lack of tunable parameters for the evaluation metrics tested, the data were not split into training and testing sets.

A comparison of correct pairwise judgments made by PyMEANT and 5-gram per-sentence BLEU is shown in Table 2. PyMEANT scores show higher correlation than BLEU on both tasks, with an especially marked difference in performance on the Chinese–English data.

	ar–en	zh–en
BLEU	37.74%	34.54%
PyMEANT	38.99%	41.18%

Table 2: Correct pairwise rankings based on PyMEANT and 5-gram per-sentence BLEU scores for the broadcast news language pairs in the DARPA GALE data set.

4 Future work

Although PyMEANT shows improved correlation with human judgments, this comes at the expense of runtime and memory costs, in particular for training the lexical similarity model and performing bipartite matching. For arguments with many words, the $O(V^2E)$ runtime of matching leads to significant slowdowns. Multiple system outputs can be scored in parallel to reduce runtime, but even on a modern computer, PyMEANT took about 20 minutes to train a lexical similarity model, then compute per-sentence scores for each of the system outputs in the testing task. BLEU scoring, on the other hand, was nearly instantaneous.

On the linguistic side, additional information may be obtainable from the argument structure of noun phrases, which PropBank does not cover. For example, the noun *ovation* and the verb *applaud* point to the same concept, and so they can be expected to have similar argument structure; however, the current PropBank-based parser would not be able to capture this equivalence. NomBank (Meyers et al., 2004) applies the PropBank model to nominal constructions as well, which may improve MEANT’s frame and argument matching performance.

References

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 178–186. Association for Computational Linguistics, August.
- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *In EACL*, pages 249–256.
- Charles J. Fillmore, Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to FrameNet. *International journal of lexicography*, 16(3):235–250.
- Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative

- graph-based parser for the Abstract Meaning Representation. In *Proceedings of ACL*. Association for Computational Linguistics.
- Chi-kiu Lo and Dekai Wu. 2013. MEANT at WMT 2013: A tunable, accurate yet inexpensive semantic frame based MT evaluation metric. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 422–428. Association for Computational Linguistics.
- Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. 2012. Fully automatic semantic MT evaluation. In *Proceedings of the 7th Workshop of Statistical Machine Translation*, pages 243–252. Association for Computational Linguistics.
- Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*, pages 24–31.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Sameer S. Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *HLT-NAACL*, pages 233–240.
- Anand Karthik Tumuluru, Chi-kiu Lo, and Dekai Wu. 2012. Accuracy and robustness in measuring the lexical similarity of semantic role fillers for automatic semantic MT evaluation. In *26th Pacific Asia Conference on Language, Information and Computation*, pages 574–581.