Enhancing BLEU with Semantic Word Vectors

John Morgan

December 19, 2014

Abstract

Machine translation metrics are proposed that attack the paucity of references problem. Semantic word vectors are incorporated into the matching algorithm for the BLEU machine translation evaluation metric.

Contents

1 Introduction

1.1 The Paucity of References Problem

Naturally occurring sentences can have billions of correct translations into one foreign language[4]. By a correct translation of a source sentence we mean any sentence in the target language that has the same meaning as the source sentence. The BLEU[6] metric was designed assuming that the system output would be compared to a set of four reference translations. In practice system developers rarely have access to more than one reference translation. In this work we assume only one human generated reference is available.

The main problem with BLEU caused by the paucity of references is that a system can produce a perfectly adequate translation and get a very low BLEU score. The designers of BLEU were aware that input sentences could have many translations into a foreign language, so they incorporated features that capture variations of references. Bleu does not require an exact match for the entire sentence. Instead, it allocates credit to sub-sentence segments by considering matches of overlapping n-grams for n equals 1 through 4. Even with this partial credit assignment and assuming four references, this coverage is not enough to be representative of billions of possible correct translations.

1.2 Improving BLEU

Making BLEU better is a goal almost as old as BLEU itself. It became clear that methods of generating synonyms and sentences with equivalent meaning would improve BLEU. Like our present project, several previous projects have been devoted to developing metrics that extend BLEU by expanding the set of references to include more sentences with equivalent meaning. Some of these projects like METEOR[1][3], TERP[8][7] and HyTER have produced popular metrics.

We list some features we would like to see in an improved BLEU metric and some features we want to keep.

- We want to assign credit to approximate matches. System output that expresses similar meaning to the input should receive adequate credit from the metric.

- We want it to adapt easily to a new language pair. That is, we want our metric to require little or no human labor for annotation, and little or no NLP resources.

- We want a metric that assigns reasonable scores to partial output. We will use this metric as a component that measures translation quality in interactive incremental language systems. Such systems need to measure the quality of partial inputs.

1.3 Word Similarities

The ultimate goal of our project is to attack the paucity of references problem directly by using a semantic vector space to generate references equivalent to a single given reference. In this paper however, we investigate an easier solution. We use the similarity measure defined in a semantic vector space generated by the tool word2vec to provide scores for $n$grams that do not have exact matches in the reference. Word2vec embeds single word tokens into a vector space, so it makes sense to measure similarity between words in that space. The word2vec developers suggest that the similarity function can be extended from words to phrases by averaging vectors. Although we do not believe that this is theoretically well-founded, we take on their suggestion and build two enhanced BLEU metrics that compute similarities between phrases. One of these metrics yields a significant boost in scores relative to BLEU. Many $n$grams that did not match exactly receive a non-zero similarity measure with reference phrases. However, an inspection of the matches indicates that a large portion of them are spurious. We conclude that it is not an appropriate use of the similarity measure to provide approximate match scores for phrases. We instead believe that more sophisticated approaches need to be taken to generate phrases with equivalent meaning to a given reference.

2 Background

In this section we describe BLEU and two other metrics with similar goals to our own.

2.1 Previous Work

2.1.1 BLEU

BLEU is still the best metric for parameter optimization[2]. It is simple, quick, inexpensive, language-independent, and correlates highly with human evaluations.

Bleu computes precision for n-gram matches for n between 1 and 4. The geometric mean of the four precisions is computed and a brevity penalty is applied if the total length of the references is greater than the total length of the system output.

2.1.2 METEOR

Meteor is an evaluation metric that takes into consideration stem, synonym, and paraphrase matches between words in addition to exact string matches. Enhancements include:

- Porter Stemming;
- Words are determined to be synonyms if they share a synset in Wordnet;

- Function word discounting.

These enhancements yielded improvements over BLEU, but they are language dependent. Later versions have incorporated methods for adapting METEOR to new languages.

METEOR extends BLEU by incorporating semantic features into the evaluation of system output. We similarly enhance BLEU with semantic knowledge.

2.1.3 TERP

TERP builds on Translation Edit Rate (TER), which is a metric that uses the minimum edit distance to score translations. TERP inherits the two stemming and synonym enhancements from METEOR. TERP takes a step up from comparing words with equivalent meaning (synonyms) to comparing phrases with equivalent meaning (paraphrases).

TERP is also grounded in human judgements. The scoring function parameters, that is, the weights associated with the diffferent edit operations, are optimized to correlate with the editing operations performed by humans.

TERP also produces as a byproduct an alignment between the hypothesis and reference.

2.1.4 HyTER

Dreyer and Marcu's work on HyTER solves the reference posity problem by employing human annotators to write alternative translations for phrases. The phrases are compiled into Recursive Transition Networks that encode an exponential number of reference sentences. It has been shown that HyTER is better at differentiating human from machine translation than BLEU, terp, and METEOR. HyTER also ranks machine translation systems performance better than these metrics, where the correct ranking is given by human judgements. HyTER also outperforms the other automatic methods as corpus size decreases from document to sentence. In summary, HyTER has clear advantages over other automatic metrics.

However, HyTER has one disadvantage we want to avoid. The process of creating the meaning equivalent networks used by HyTER requires a prohibitively large amount of human labor.

2.2 Word2vec

The word2vec tool is used to build a vector space word distribution model of the target language. The cosine similarity measure is used to assign scores to words in the reference transcriptions.

Word2vec only has one language-dependent requirement:a large monolingual corpus of text. It does not require other NLP resources like a parser, a stemmer, or a part of speech tagger.

The representations of words are learned using the distributed Skip-gram or Continuous Bag-of-Words (CBOW) models[5]. These models learn word representations using a neural network architecture that predicts the neighbors of a word.

3 Methods

Before we describe the methods we used in this work, we give a classification of the approaches that could be used to attack the paucity of references problem. We group the approaches into 2 major classes: segment methods and hierarchical methods. The hierarchical methods use a parse tree to assemble word vectors into phrase vectors and phrase vectors into sentence vectors. We will not deal with hierarchical approaches in this paper. We subdivide the segment approaches into word-level methods and phrase-level methods. Each of these approaches can be further classified into two more classes depending on how we use the vector space embedding. There are two features of semantic vector space embeddings that we use to design our methods. The vector space embeddings come with an associated similarity measure. One class of approaches assigns a similarity to references. The methods that use this feature we will call similarity methods. The similarity measure associated with vector space embeddings can also be used to generate a list of vectors in a neighborhood of a given vector. The vectors in the neighborhood can be ranked by their similarity to the given vector. We use this feature to generate extra references for a given reference. Methods that use this feature we will call generation methods. In this paper we only report on segment similarity methods

We single out three similarity methods and we name them SLEU, VLEU, and XLEU for scalar, vector, and cross BLEU. SLEU is a word similarity metric. VLEU and XLEU are phrase similarity metrics.

The BLEU algorithm loops over the ngrams in the system output and the reference transcriptions. A match count is accumulated for each of the four precision computations. A 1 is added to the count if the system and reference n-grams match and a 0 is added otherwise. All three methods, SLEU, VLEU, and XLEU compute the similarity between the system output and reference $n$grams and adds this number instead of a 1 or a 0 to the match count used in the precision calculation. SLEU only adds similarity scores for 1-grams. VLEU adds similarity scores for 1 to 1, 2 to 2, 3 to 3, and 4 to 4-grams. XLEU adds similarity scores for 1 to 1, 2 , 3, and 4-grams, 2 to 1, 2, 3, and 4-grams, 3 to 1, 2, 3, and 4- grams, and 4 to 1, 2, 3, and 4-grams. In the XLEU case, we compute similarities with the remaining reference n-grams. That is, we do not consider the similarity between non-exact candidate j-grams and reference k-grams that have already been matched exactly. Unless the candidate j-gram occurrs several times in the reference. Note that – as in the case for scalar and vector BLEU – each time a candidate j-gram matches a reference k-gram, the count of reference k-grams is decremented by its similarity to the candidate j-gram. If this count falls below 0 it is removed from the references.

Our simplest method, SLEU, only differs from BLEU in the precision computation for 1grams. When a candidate 1-gram does not match any 1-gram in

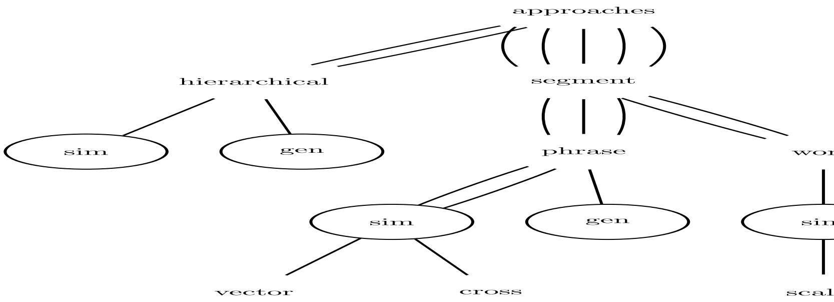

Figure 1: Possible approaches for incorporating semantic vectors into an evaluation metric.

the reference, we compute the similarities with each of the remaining 1-grams in the reference. Instead of adding 0 to our match count, we add the maximum of these similarities. We then remove one instance of the 1-gram that yielded the maximum similarity. This step of removing the argmax 1-gram is done to mitigate the overgeneration problem. In the discussion section we consider a problem that this step causes and for which we do not yet have a solution.

We consider the 1-gram case separately, because it has a stronger theoretical basis. The similarities are given to us by word2vec. We used the pre-trained vectors available for download on the word2vec webpage. These are 300 dimensional vectors. The vector space embedding performed by word2vecd operates on 1-grams. Similarities for higher n-grams are computed and we exploit this in our phrase-level methods, VLEU and XLEU, but we see this as having weaker theoretical foundations.

VLEU is similar to SLEU; when a system output n-gram does not match exactly with any reference n-gram, its similarity with the remaining n-grams is computed and the maximum of these similarities is added to the match count. XLEU computes yet more similarities. The similarity measure that operates on the embedded word vectors allows the similarity between j-grams and k-grams for j different from k to be computed. For a non-exact matching j-gram we compute similarities with k-grams for k between 1 and 4. We add the maximum of these similarities to our match count.

Note that when computing precision, we need to divide by the total number of possible matches. What is that number in the XLEU case? When computing j-BLEU or j-precision, All the $n$grams have a chance of matching or being similar to all the j-grams not only the j-grams. So the denominator in the precision formula should be a count of the total number of n-grams. This is different from the j-BLEU case. When computing j-BLEU the denominator is a count of only the j-grams. This makes a large difference in the final score.

3.1 Data

We tested the metrics on astandard corpus and an artificial corpus. After running the metrics on the WMT13 corpus, we found many spurious n-grams getting credit from the similarity score. For example, we found that many non-exact matching candidate n-grams would achieve maximum similarity with the 1-gram “the”. In order to investigate the cause of these problems, we wrote a script that generates sentences from a grammar. We made the leaves of the grammar be very similar so that we could generate sentence with similar meaning. For example, for the noun constituent we used the words “woman”, “girl”, “person”, “boy”, and “man”. After generating a llist of sentences, we split the list in two and pasted the two lists into a parallel corpus.

3.1.1 Standard Corpora

The WMT13 German to English corpus consists of 3000 segment pairs. Each pair consists of a system output and one reference. The system output has

55786 tokens, the reference has 56089 tokens.

3.1.2 Artificial Corpus

We generated a small corpus of sentence pairs with simple structure. We wantted to create a corpus of segment pairs that were very close in meaning.

candidate	reference
the person went by the supermarket	a person ran to a store
the girl ran by the store	we drove by the store
the woman drove to a supermarket	a girl went around the store
the girl drove to a mall	we went around the supermarket

Table 1: Some examples of sentences from an artificial corpus.

3.2 Similarity Thresholding

The results of running the metrics on the artificial corpus indicated that high similarity values were associated with few spurious approximate matches. As similarity values decreased, more spurious approximate matches achieved the maximum similarity. This observation led us to investigate the use of a similarity threshold. If the similarity score of a candidate reference n-gram pair is above a fixed number we consider it an approximate match and add it to the match count, otherwise we ignore it.

4 Results

Table 2 shows that the total number of exact and approximate matched n-grams is very similar for the BLEU, SLEU, and VLEU metrics. The totals for the XLEU metric are much largere.

n-gram matches				precisions				
1	2	3	4	1	2	3	4	
BLEU	27867	13406	7070	3853	0.50	0.25	0.14	0.08
SLEU	27939.7	13406	7070	3853	0.50	0.25	0.14	0.08
VLEU	27938	13489.3	7151.3	3912.4	0.50	0.25	0.14	0.08
XLEU	48570.63	47512.57	39648.82	30769.48	0.30	0.30	0.25	0.19

Table 2: Matches and precisions on WMT13.

4.1 Qualitative Results

5 Discussion

Bleu is a precision metric. Consider the following candidate/reference pair:

metric	1-grams	2-grams	3-grams	4-grams
BLEU SLEU VLEU	56089	53089	50105	47138
XLEU	159283	159267	159216	159060

Table 3: Total n-grams in the WMT13 corpus.

sleu	1227.68	238.0	62.0	10.0
vleu	1227.68	397.09	182.42	85.73

Table 4: Total number of n-grams in artificial corpus.

the the the the the the the
the cat on the mat

We call this the the overgeneration problem (the two the tokens are not a typo) We handle this problem by removing a the in the reference each time it is matched . This takes care of the the overgeneration problem in the standard BLEU metric. The the overgeneration problem remains in our method. Consider the candidate reference pair: The price for the weapon I learned there I found out the price of the weapon only there The word “for” does not appear in the reference. Our method obtains similarity scores between “for” and each word in the reference that has not been mathced yet. Unfortunately, the word with highest similarity to “for” is “there. “there” is chosen as the match for “for” and is removed from the referenced word list. When we later try to find a match for “there” an exact match is not made. In this case the standard BLEU metric assigns a higher score to the candidate than our method.

The xleu metric assigns a perfect match to the pair of 4-grams in table 7.

This is a problem. The metric is assigning perfect scores to n-grams that are badly oredered. If the system and reference have the same words in different oreders a perfect score is assigned to the n-gram. This happens because the vector representation of the 4-gram is taken as the average of each of its component 1-grams.

The pair in table 8 is an example of a good score assignment by XLEU.

metric	WMT13	artificial
BLEU	0.1940	0.0498
SLEU	0.194107449678	0.0498
VLEU	0.1957	0.0779

Table 5: Scores.

metric	score thresholds		
	0.90	0.80	0.70
SLEU	0.1940		
VLEU	0.1941		
xleu	0.2074		

Table 6: Scores by thresholdon WMT13.

References

[1] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL*. Association for Computational Linguistics, 2005.

[2] Daniel Cer, Christopher D. Manning, and Daniel Jurafsky. The best lexical metric for phrase-based statistical mt system optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 555–563, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[3] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

[4] Markus Dreyer and Daniel Marcu. 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*, pages 162–171, Montréal, Canada, June 2012. Association for Computational Linguistics.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *In Proceedings of Workshop at ICLR, 2013*, 2013.

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[7] Matt Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. TERp: A system description. In *Proceedings of the First NIST Metrics for Machine Translation Challenge (MetricsMATR)*, October 2008.

candidate	reference	similarity
the woman drove around	woman drove around the	1.0

Table 7: XLEU assigns a perfect match to this pair of 4-grams.

drove around the store	went around the store	0.93

Table 8: A good score assignment by XLEU.

[8] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, 2006.