Representation Based Translation Evaluation Metrics

Boxing Chen and Hongyu Guo
National Research Council Canada
first.last@nrc-cnrc.gc.ca

Abstract

Precisely evaluating the quality of a translation against human references is a challenging task due to the flexible word ordering of a sentence and the existence of a large number of synonyms for words. This paper proposes to evaluate translations with distributed representations of words and sentences. We study several metrics based on word and sentence representations and their combination. Experiments on the WMT metric task shows that the metric based on the combined representations achieves the best performance, outperforming the state-of-the-art translation metrics by a large margin. In particular, training the distributed representations only needs a reasonable amount of monolingual, unlabeled data that is not necessary drawn from the test domain.

1 Introduction

Automatic machine translation (MT) evaluation metrics measure the quality of the translations against human references. They allow rapid comparisons between different systems and enable the tuning of parameter values during system training. Many machine translation metrics have been proposed in recent years, such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), TER (Snover et al., 2006), Meteor (Banerjee and Lavie, 2005) and its extensions, and the MEANT family (Lo and Wu, 2011), amongst others.

Precisely evaluating translation, however, is not easy. This is mainly caused by the flexible word ordering and the existence of the large number of synonyms for words. One straightforward solution to improve the evaluation quality is to increase the number of various references. Nevertheless, it is expensive to create multiple references. In order to catch synonym matches between the translations and references, synonym dictionaries or paraphrasing tables have been used. For example, Meteor (Banerjee and Lavie, 2005) uses WordNet (Miller, 1995); TER-Plus (Snover et al., 2009) and Meteor Universal (Denkowski and Lavie, 2014) deploy paraphrasing tables. These dictionaries have helped to improve the accuracy of the evaluation; however, not all languages have synonym dictionaries or paraphrasing tables, especially for those low resource languages.

This paper leverages recent developments on distributed representations to address the above mentioned two challenges. A distributed representation maps each word or sentence to a continuous, low dimensional space, where words or sentences having similar syntactic and semantic properties are close to one another (Bengio et al., 2003; Socher et al., 2011; Socher et al., 2013; Mikolov et al., 2013). For example, the words *vacation* and *holiday* are close to each other in the vector space, but both are far from the word *business* in that space.

We propose to evaluate the translations with different word and sentence representations. Specifically, we investigate the use of three widely deployed representations: one-hot representations, distributed word representations learned from a neural network model, and distributed sentence representations computed with recursive autoencoder. In particular, to leverage the different advantages and focuses, in terms of benefiting evaluation, of various representations, we concatenate the three representations to form one vector representation for each sentence. Our experiments on the WMT metric task show that the metric based on the concatenated representation outperforms several state-of-the-art machine translation metrics, by a large margin on both segment and system-level. Furthermore, our results also indicate that the representation based metrics are robust to a variety of training conditions, such as the data volume and domain.

*Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 150–155,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

2 Representations

A representation, in the context of NLP, is a mathematical object associated with each word, sentence, or document. This object is typically a vector where each element's value describes, to some degree, the semantic or syntactic properties of the associated word, sentence, or document. Using word or phrase representations as extra features has been proven to be an effective and simple way to improve the predictive performance of an NLP system (Turian et al., 2010; Cherry and Guo, 2015). Our evaluation metrics are based on three widely used representations, as discussed next.

2.1 One-hot Representations

Conventionally, a word is represented by a one-hot vector. In a one-hot representation, a vocabulary is first defined, and then each word in the vocabulary is assigned a symbolic ID. In this scenario, for each word, the feature vector has the same length as the size of the vocabulary, and only one dimension that corresponds to the word is on, such as a vector with one element set to 1 and all others set to 0. This feature representation has been traditionally used for many NLP systems. On the other hand, recent years have witnessed that simply plugging in distributed word vectors as real-valued features is an effective way to improve a NLP system (Turian et al., 2010).

2.2 Distributed Word Representations

Distributed word representations, also called word embeddings, map each word deterministically to a real-valued, dense vector (Bengio et al., 2003). A widely used approach for generating useful word vectors is developed by (Mikolov et al., 2013). This method scales very well to very large training corpora. Their skip-gram model, which we adopt here, learns word vectors that are good at predicting the words in a context window surrounding it. A very promising perspective of such distributed representation is that words that have similar contexts, and therefore similar syntactic and semantic properties, will tend to be near one another in the low-dimensional vector space.

2.3 Sentence Vector Representations

Word level representation often cannot properly capture more complex linguistic phenomena in a sentence or multi-word phrase. Therefore, we adopt an effective and efficient method for multi-word phrase distributed representation, namely the greedy unsupervised recursive auto-encoder strategy (RAE) (Socher et al., 2011). This method works under an unsupervised setting. In particular, it does not rely on a parsing tree structure in order to generate sentence level vectors. This characteristic makes it very desirable for applying it to the outputs of machine translation systems. This is because the outputs of translation systems are often not syntactically correct sentences; parsing them is possible to introduce unexpected noise.

For a given sentence, the greedy unsupervised RAE greedily searches a pair of words that results in minimal reconstruction error by an auto-encoder. The corresponding hidden vector of the auto-encoder (denoted as the two children's parent vector), which has the same size as that of the two child vectors, is then used to replace the two children vectors. This process repeats and treats the new parent vector like any other word vectors. In such a recursive manner, the parent vector generated from the word pool with only two vectors left will be used as the vector representation for the whole sentence. Interested readers are referred to (Socher et al., 2011) for detailed discussions of the strategy.

2.4 Combined Representations

Each of the above mentioned representations has a different strength in terms of encoding syntactic and semantic contextual information for a given sentence. Specifically, the one-hot representation is able to reflect the particular words that occur in the sentence. The word embeddings can recognize synonyms of words appearing in the sentence, through the co-occurrence information encoded in the vector's representation. Finally, the RAE vector can encode the composed semantic information of the given sentence. These observations suggest that it is beneficial to take various types of representations into account.

The most straightforward way to integrate multiple vectors is using concatenation. In our studies here, we first compute the sentence-level one-hot, word embedding, and RAE representations. Next, we concatenate the three sentence-level representations to form one vector for each sentence.

3 Representations Based Metrics

Our translation evaluation metrics are built on the four representations as discussed in Section 2.

Consider we have the sentence representations for the translations (t) and references (r), the translation quality is measured with a similarity score computed with Cosine function and a length penalty. Suppose the size of the vector is N, we calculate the quality as follows.

$$\text{Score}(t, r) = \text{Cos}^{\alpha}(t, r) \times P_{len} \tag{1}$$

$$\text{Cos}(t, r) = \frac{\sum_{i=1}^{i=N} v_i(t) \cdot v_i(r)}{\sqrt{\sum_{i=1}^{i=N} v_i^2(t)}\sqrt{\sum_{i=1}^{i=N} v_i^2(r)}} \tag{2}$$

$$P_{len} = \begin{cases} exp(1 - l_r/l_t) & \text{if } (l_t < l_r) \\ exp(1 - l_t/l_r) & \text{if } (l_t \geq l_r) \end{cases} \tag{3}$$

where α is a free parameter, $v_i(.)$ is the value of the vector element, P_{len} is the length penalty, and l_r, l_t are length of the translation and reference, respectively.

In the scenarios of there exist multiple references, we compute the score with each reference, then choose the highest one. Also, we treat the document-level score as the weighted average of sentence-level scores, with the weights being the reference lengths, as follows.

$$\text{Score}_d = \frac{\sum_{i=1}^{D} \text{len}(r_i)\text{Score}_i}{\sum_{i=1}^{D} \text{len}(r_i)} \tag{4}$$

where Score$_i$ denotes the score of sentence i, and D is the size of the document in sentences. With these score equations, we then can formulate our five presentations based metrics as follows.

For the one-hot representation metric, once we have the representations of the words and n-grams, we sum all the vectors to obtain the representation of the sentence. For efficiency, we only keep the entries which are not both zero in the reference and translation vectors. After we generate the two vectors for both translation and reference, we then compute the score using Equation 1.

For the word embedding based metric, we first learn the word vector representation using the code provided by (Mikolov et al., 2013) [1]. Next, following (Zou et al., 2013), we average the word embeddings of all words in the sentence to obtain the representation of the sentence.

As discussed in Section 2.4, the three sentence-level one-hot, word embedding and RAE representations have different strength when they are used to compare two sentences. In our metric here, each of the three vectors is first scaled with a particular weight (learned on dev data) and then the vectors are concatenated. With these concatenation vectors, we then calculate the similarity score using Equation 1.

For comparison, we also combine the strength of the three representations using weighted average of the three metrics computed. Weights are tuned using development data.

4 Experiments

We conducted experiments on the WMT metric task data. Development sets include WMT 2011 all-to-English, and English-to-all submissions. Test sets contain WMT 2012, and WMT 2013 all-to-English, plus 2012, 2013 English-to-all submissions. The languages "all" include French, Spanish, German and Czech. For training the word embedding and recursive auto-encoder model, we used WMT 2013 training data [2].

We compared our metrics with smoothed BLEU (mteval-v13a), TER [3], Meteor v1.0 [4], and Meteor Universal (i.e. v1.5) [5]. We used the default settings for all these four metrics.

When considering the representation based metrics, we tuned all the parameters to maximize the system-level γ score for all representation based metrics on the dev sets. We tuned the weights for combining the three vectors automatically, using the downhill simplex method as described in (Press et al., 2002). The weights are 1 for the RAE vector, about 0.1 for the word embedding vector, and around 0.01 for the one-hot vector, respectively. We tuned other parameters manually. Specifically, we set n equal to 2 for the one-hot n-gram representation, the vector size of the recursive auto-encoder to 10, and the vector size of word embeddings to 80.

Following WMT 2013's metric task (Macháček and Bojar, 2013), to measure the correlation with human judgment, we use Kendall's rank correlation coefficient τ for the segment level, and Pearson's correlation coefficient (γ in the below tables and figures) for the system-level respectively.

[1] https://code.google.com/p/word2vec/

[2] http://www.statmt.org/wmt13/translation-task.html

[3] http://www.cs.umd.edu/ snover/tercom/

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

[5] Meteor universal package does not include paraphrasing table for other target language except English, so we did not run Out-of-English experiments for this metric.

	Into-Eng		Out-of-Eng	
metric	seg τ	sys γ	seg τ	sys γ
BLEU	0.220	0.751	0.179	0.736
TER	0.211	0.742	0.175	0.745
Meteor	0.228	0.824	0.180	0.778
Met. Uni.	0.249	0.808	–	–
One-hot	0.235	0.795	0.183	0.773
Word emb.	0.212	0.818	0.175	0.788
RAE vec.	0.203	0.856	0.171	0.780
Comb. rep.	**0.259**	**0.874**	**0.191**	**0.832**
Wghted avg.	0.247	0.863	0.185	0.798

Table 1: Correlations with human judgment on WMT data for Into-English and Out-of-English task. Results are averaged on all test sets.

4.1 General Performance

We first report the main experimental results conducted on the Into-English and Out-of-English tasks. Results in Tables 1 suggest that metrics based on three single representations all obtained comparable or better performance than BLEU, TER and Meteor. In particular, the metric based on recursive auto-encoder outperformed the other testing metrics on system-level. When combining the strengths of the three representations, our experimental results show that the metric based on the combined representation outperformed all state-of-the-art metrics by a large margin on both segment- and system-level.

Regarding the evaluation speed of the representation metrics, it took around 1 minute to score about 2000 sentences with the above settings on a machine with a 2.33GHz Intel CPU. It is worth noting that if we increase the vector size of the RAE model and word embeddings, longer execution time is expected for the scoring processes.

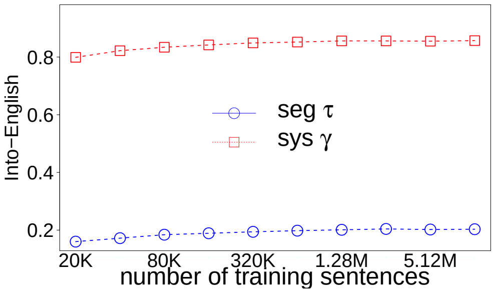

Figure 1: Correlations with human judgment on WMT data for Into-English task for combined representation based metric when increasing the size of the training data.

4.2 Effect of the Training Data Size

In our second experiment, we measure the performance on the Into-English task and increase the training data from 20K sentences to 11 million sentences. The sentences are randomly selected from the whole training data, which include the English side of WMT 2013 French-to-English parallel data (“Europarl v7”, “News Commentary” and “UN Corpus”). The results are reported in Figure 1. From this figure, one can conclude that the performance improves with the increasing of the training data, however, when more than 1.28M sentences are used, the performance stabilizes. This result indicates that training a stable and good model for our metric does not need a huge amount of training data.

4.3 Sensitivity to Data Across Domains

The last experiment aimed at the following question: should the test domain be consistent with the training domain? In this experiment, we sampled three training sets from different domain data sets in equal number (136K) of sentences: Europarl (EP), News Commentary (NC), and United Nation proceedings (UN), while the test domain remains the same, i.e., the news domain. The metric trained on NC domain data achieved slightly higher segment-level τ score (0.181 vs 0.178 for EP, 0.176 for UN) and system-level Pearson’s correlation score γ (0.821 vs 0.820 for EP, 0.817 for UN). Nevertheless, the results are consistent across domains. This is explainable: although the same test sentence may have different representations w.r.t. the training domain, the distance between the translation and its reference may stay consistent. Practically, the training and test data not necessary being in the same domain is a very attractive characteristic for the translation metrics. It means that we do not have to train the word embeddings and RAE model for each testing domain.

4.4 Cope with Word Ordering and Synonym

In order to better understand why metrics based on combined representations can achieve better correlation with human judgment than other metrics, we select, in Table 2, some interesting examples for further analysis.

Consider, for instance, the first reference (denoted as “1 R” in Table 2) and their translations. If we replace the word *vacation* in the reference with words *business* and *holiday*, respectively, then we

id	sentence	BLEU	rep.
1 R	i had a wonderful vacation in italy	–	–
1 H1	i had a wonderful business in italy	0.489	0.555
1 H2	i had a wonderful holiday in italy	0.489	0.865
1 H3	in italy i had a wonderful vacation	0.707	0.804
1 H4	vacation in i had a wonderful italy	0.508	0.305
2 R	but the decision was not his to make	–	–
2 H1	but it is not up to him to decide	0.063	0.652
2 H2	but the decision not him to take	0.241	0.620
2 H3	but the decision was not the to make	0.595	0.612
3 R	they were set to go on trial in jan	–	–
3 H1	they should appear in court in jan	0.109	0.498
3 H2	the trial was scheduled in jan	0.109	0.454
3 H3	the procedures were prepared in jan	0.109	0.445

Table 2: Examples evaluated with smoothed BLEU and combined representation based metric. Examples 2-3 are picked up from the real test sets; human judgment ranks H1 better than H2, and H2 better than H3 for each of these example sentences. The combined representation based metric better matches human judgment than BLEU does.

have hypothesis 1 and hypothesis 2, denoted as "1 H1" and "1 H2", respectively, in Table 2 . In this scenario, the metric BLEU assigns the same score of 0.489 for these two translations. In contrast, the representation based metric associates hypothesis 2 with a much higher score than that of hypothesis 1, namely 0.865 and 0.555, respectively. In other words, the score for hypothesis 2 is close to one, suggesting that the RAE based metric considers this translation is almost identical to the reference. The reason here is that the vector representations for the two words are very near to one another in the vector space. Consequently, the representation based metric treats the *holiday* as a synonym of *vacation*, which matches human's judgment perfectly.

Let us continue with this example. Suppose, in hypothesis 3, we reorder the phrase *in italy*. The representation based metric still considers this to be a good translation with respect to the reference, thus associating a very close score as that of the reference, namely 0.804. The reason for representation metric's correct judgment is that H3 and the reference, in the vector space, embed very similar semantic knowledge, although they have different word orderings. Now let us take this example a bit further. We randomly mess up the words in the reference, resulting in hypothesis 4 (denoted as "1 H4" as shown in Table 2). In such scenario, the representation metric score drops sharply because the syntactic and semantic information embedded in the vector space is very different from the reference. Interestingly, the BLEU metric still consider this translation is not a very bad translation.

We made up the first example sentence for illustrative purpose, however, the examples 2-3 are picked up from the real test sets. According to the human judgment, hypothesis 1 (H1) is better than hypothesis 2 (H2); hypothesis 2 is better than hypothesis 3 (H3) for each of these example sentences. These results indicate that the combined representation based metric better matches the human judgment than BLEU does.

5 Conclusion

We studied a series of translation evaluation metrics based on three widely used representations. Experiments on the WMT metric task indicate that the representation metrics obtain better correlations with human judgment on both system-level and segment-level, compared to popular translation evaluation metrics such as BLEU, Meteor, Meteor Universal, and TER. Also, the representation-based metrics use only monolingual, unlabeled data for training; such data are easy to obtain. Furthermore, the proposed metrics are robust to various training conditions, such as the data size and domain.

Acknowledgements

The authors would like to thank Colin Cherry and Roland Kuhn for useful discussions.

References

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.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for twitter named entity recognition. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

G. Doddington. 2002. Authomatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the Human Language Technology Conference*, page 128132, San Diego, CA.

Chi-kiu Lo and Dekai Wu. 2011. Meant: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 220–229, Portland, Oregon, USA, 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.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

George A. Miller. 1995. Wordnet: A lexical database for english. *Comunications of the ACM*, 38:39–41.

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 (ACL)*, pages 311–318, Philadelphia, July. ACL.

W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. 2002. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK.

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 Association for Machine Translation in the Americas*.

Matthew G. Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Ter-plus: Paraphrase, semantic, and alignment enhancements to translation edit rate. In *Machine Translation*, volume 23, pages 117–127.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*, pages 384–394.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA, October. Association for Computational Linguistics.