The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Impact of a New Word Embedding Cost Function on Farsi-Spanish Low-Resource Neural Machine Translation

Benyamin Ahmadnia,[1] **Bonnie J. Dorr**[2]
[1]Department of Computer Science, Tulane University, New Orleans, LA, USA
[2]Institute for Human and Machine Cognition (IHMC), Ocala, FL, USA
ahmadnia@tulane.edu, bdorr@ihmc.us

Abstract

Neural Machine Translation (NMT) relies heavily on word embeddings, which are continuous representations of words in a vector space, obtained from large monolingual data and, independently, from bilingual data for NMT model training. Word embeddings have proven to be invaluable for performance improvements in natural language analysis tasks that otherwise suffer from data scarcity. This paper defines a new cost function—demonstrated on Farsi-Spanish low-resource attention-based NMT—that encodes word similarity as distances within a word embedding space. The novelty of this cost function is that it encourages our attentional NMT model to generate words that are close to their references in the embedding space. This approach encourages the decoder to select acceptably similar words when potential candidates are found to be Out-Of-Vocabulary (OOV). Experimental results demonstrate improvements of our attentional NMT model over a community-standard NMT baseline model.

Introduction

Recent years have witnessed considerable improvements in Neural Machine Translation (NMT) performance based on encoder-decoder architectures (Sequence-to-Sequence). NMT exploits Convolutional Neural Networks (CNNs) (Gehring et al. 2017), Recurrent Neural Networks (RNNs) (Sutskever, Vinyals, and Le 2014; Cho et al. 2014; Wu et al. 2016), or Transformers (Vaswani et al. 2017) to learn mappings between source sentences and their corresponding target translations. In addition, attention-based mechanisms (Bahdanau, Cho, and Bengio 2015; Luong, Pham, and Manning 2015) help soft-align the encoded source words with the predictions, further improving the translation.

NMT performance suffers in low-resource conditions where sufficient parallel texts cannot be obtained. The computational cost of the output layer in NMT systems increases with the target language vocabulary size. One might consider limiting vocabulary size, ignoring low-frequency words to reduce computational costs, but this decreases translation quality due to the large number of Out-Of-Vocabulary (OOV) words. This trade-off is frequently addressed by using a cost function that supports NMT model training, e.g., softmax cross-entropy (Chousa, Sudoh, and Nakamora 2018). Consider the optimization of parameters to generate a reference word *say*. The probability of *say* is close to 1, whereas the probabilities of other words are closer to 0, regardless of their meanings. This is true even for words similar to *say*, such as *tell*, i.e., an occurrence of *tell* is penalized as heavily as much more dissimilar words.

We hypothesize that attentional NMT performance substantially improves through the introduction of a new *word embedding cost* function that assigns a smaller penalty to similar words than to dissimilar words, e.g., assignment of a lower penalty to *tell*, for a reference word *say*, than to dissimilar words such as *look*. To date, the performance of NMT systems relies heavily on availability of large, parallel, domain-specific data. Because translation of OOV words is central to the success of NMT, low-resource languages are particularly challenging due to the lack of advanced linguistic tools and extreme sparcity of parallel training data.

The impact of an improved word-embedding cost function is high when one considers OOV words that emerge under such sparse training-data conditions. The typical approach is to optimize parameters by penalizing occurrences of other words, and to produce a special OOV symbol as part of the output. Our approach overcomes adverse effects associated with production of uninterpretable OOV symbols, enabling generation of similar words in place of such symbols. Our cost function encourages the NMT decoder to generate words close to their references in the embedding space; this helps the decoder to choose similar acceptable words when the best candidates are not included in the vocabulary.

In summary, our work is designed to address the low-resource NMT bottleneck, where defining a large vocabulary is nearly impossible. We undertake side-by-side analyses with state-of-the-art approaches that have comparably small vocabulary sizes. Results suggest our method works well against systems with similarly limited vocabulary—on the order of 1K, a standard size for low-resource conditions (Chousa, Sudoh, and Nakamora 2018).

The next section presents language issues associated with our case study. The relevant mathematical bases for our attention NMT is then presented, followed by a description of our word embedding cost function, experiments and results. We discuss the upshot of our experiments and contrast other approaches to our own, and then conclude with future work.

Language Issues

Below we highlight issues specific to languages that exacerbate the low-resource problem.

Farsi Language Issues

Farsi suffers from a shortage of digitally available parallel and monolingual texts. Many characteristics of Farsi are shared only by Arabic, so it is difficult to leverage linguistic knowledge about other languages to fill the data gap. Farsi makes no use of articles (*a*, *an*, *the*) and does not distinguish between capital and lower-case letters. Symbols and abbreviations are rarely used. Farsi is written in Arabic script, so it uses diacritic marks to indicate vowels, which are generally omitted except in infant writing or in texts for language learning. Sentence structure is different from that of English, with parts of speech (e.g., nouns, subjects, adverbs and verbs) placed in different locations in the sentence, or even omitted. Farsi words may have many different accepted spellings, and translators may invent new words, thus yielding Out-Of-Vocabulary (OOV) words.

Spanish Language Issues

Spanish is a high-resource language, but still differs from other languages enough to make it difficult to leverage linguistic knowledge to fill data gaps. Spanish language punctuation is very close to, but not the same as, that of English. For example, in Spanish, exclamation and interrogative sentences are preceded by inverted question and exclamation marks. Also, in a Spanish conversation, a change in speakers is indicated by a dash, whereas in English, each speaker's remark is placed in separate paragraphs. Formal and informal translations diverge considerably. Unlike English, inflection, declination and grammatical gender are important features, and it is quite common to drop the subject of a sentence.

Farsi-Spanish Divergences

A number of *divergences* (Dorr 1994; Dorr et al. 2002) between low-resource (e.g., Farsi) and high-resource (e.g., Spanish) languages pose many translation challenges. In Farsi, the modifier precedes the word it modifies, and in Spanish the modifier follows the head word (although it may precede the head word under certain conditions). In Farsi, the sentences follow a "Subject", "Object", "Verb" (SOV) order, and in Spanish, the sentences follow the "Subject", "Verb", "Object" (SVO) order (Ahmadnia, Serrano, and Haffari 2017). Such distinctions are exceedingly prevalent and thus pose many challenges for machine translation.

Attention-based Neural Machine Translation

Following (Bahdanau, Cho, and Bengio 2015), we adopt *attention-based* an encoder-decoder that selectively focuses on sub-parts of the sentence during translation. An encoder transforms a source sentence $x = x_1, x_2, ..., x_J$ to an internal representation $h = h_1, h_2, ..., h_J$. A a decoder transforms h to the target sentences $y = y_1, y_2, ..., y_I$. Source-to-target translation is achieved by finding the best target language sentence $\hat{y}$ that maximizes the conditional probability:

$$\hat{y} = \arg\max_{y} P(y|x) \tag{1}$$

Conditional probability of the target sentence is:

$$P(y|x) = \prod_{i=1}^{I} P(y_i|y_{<i}, x) \tag{2}$$

We adopt a standard implementation of encoder/decoder as Recurrent Neural Networks (RNNs). The encoder converts source words into a sequence of vectors, and the decoder generates target words one-by-one based on the conditional probability shown (Equation 2). Specifically, the encoder takes a sequence of source words as inputs and returns forward hidden vectors $\overrightarrow{h_j}(1 \leq j \leq J)$ of the forward-RNN:

$$\overrightarrow{h_j} = f(\overrightarrow{h_{j-1}}, x) \tag{3}$$

Similarly, backward hidden vectors $\overleftarrow{h_j}(1 \leq j \leq J)$ of the backward-RNN are obtained, in the reverse order.

$$\overleftarrow{h_j} = f(\overleftarrow{h_{j-1}}, x) \tag{4}$$

These forward and backward vectors are concatenated to make source vectors $h_j(1 \leq j \leq J)$ based on Equation (5):

$$h_j = \left[\overrightarrow{h_j}; \overleftarrow{h_j}\right] \tag{5}$$

The decoder takes source vectors as inputs and returns target words, starting with the initial hidden vector h_J (concatenated source vector). Target words are generated in a recurrent manner using the decoder's hidden state and an output context. The conditional output probability of a target language word y_i is defined as follows:

$$P(y_i|y_{<i}, x) = softmax\left(f(d_i, y_{i-1}, c_i)\right) \tag{6}$$

where f is a non-linear function and d_i is a the hidden state of the decoder at step i:

$$d_i = g(d_{i-1}, y_{i-1}, c_i) \tag{7}$$

Here, g is a non-linear function that takes its previous state vector and previous output word as inputs and updates its state vector. c_i is a context vector to retrieve source inputs in the form of a weighted sum of the source vectors h_j, first taking as input the hidden state d_i at the top layer of a stacking Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997).

The goal is to derive a context vector c_i that captures relevant source information and enables prediction of the current target word y_i. While s variety of models may be used to derive a range of different context vectors c_i, our choice for c_i is formulated as follows:

$$c_i = \sum_{j=1}^{J} \alpha_{t,j} h_j \tag{8}$$

where h_j is the annotation of source word x_j and $\alpha_{t,j}$ is a weight for the j^{th} source vector at time step t to generate y_i:

$$\alpha_{t,j} = \frac{exp\left(score\left(d_i, h_j\right)\right)}{\sum_{j'=1}^{J} exp\left(score\left(d_i, h_{j'}\right)\right)} \tag{9}$$

The score function above may be defined in variety of ways as discussed by (Luong, Pham, and Manning 2015). We use *dot* attention for this score function calculated as follows:

$$score(d_i, h_j) = d_i^T h_j \tag{10}$$

This scalar product score ensures that the decoder puts more weight (attention) on source vectors that are close to its state vector d_i.

Word Embedding Cost Function

Word embeddings—continuous representations of words in a vector space—are obtained for NMT by leveraging a large monolingual dataset and (independently) a bilingual dataset. To learn meaningful word embeddings, one must devise a cost function that is to be minimized. This section presents a new cost function for Farsi-Spanish low-resource NMT model that assigns a small penalty to word pairs according to their degree of similarity (i.e., a small penalty for similar words and a large penalty to dissimilar words).

This cost function is defined as a weighted average of distances between word vectors of a reference word and others in the target language vocabulary; the weights are given by generation probabilities in the *softmax* layer (Chousa, Sudoh, and Nakamora 2018). Thus, the cost function explicitly penalizes word generation of dissimilar words with high probabilities and also encourages similar words to have high probabilities. This is beneficial in low-resource situations where many words emerge as OOVs in translation outputs. NMT models optimized by this cost function enable generation of similar in-vocabulary words, thus avoiding standard back-off approach of generating special OOV symbols.

Softmax cross-entropy cost is a commonly used cost function for multi-label classification and NMT word generation:

$$C_{ent} = -\sum_{i=1}^{I}\sum_{j=1}^{J} y_{ij} \log P_\theta(y_{ij}|y_{<i}, X) \tag{11}$$

where y_{ij} is j^{th} element of the one-hot vector corresponding to i^{th} words of the target sentence.

However, this standard cost function penalizes all words other than the reference word, even for similar words. To avoid this issue, a new cost function, which we call *word embedding cost*, is defined as a weighted average of distances to a reference word in continuous vector space:

$$C_{emb} = \sum_{i=0}^{I}\sum_{n=0}^{N} P(y_i|y_{<i}, X) d[E(V_n)), E(y_i)] \tag{12}$$

where V_n is n^{th} word in the target language vocabulary, E notation refers to embedding, $E(y_i)$ denotes a vector of word y_i, and d calculates the distance between two word vectors:

$$d(s,t) = ||s - t|| \tag{13}$$

Word embedding cost is defined as a weighted average of distances to a reference word in continuous vector space. Weights are based on word output probabilities in Equation (6). Vectors of entire vocabulary are compared to each vector. Code-execution took one week, comparable to that of other approaches that use a cost function (Ahmadnia, Kordjamshidi, and Haffari 2018). This approach reduces <*UNK*> tokens in translation, the outcome of which is trained by this approach to produce explicit <*UNK*> tokens.

Experiments and Results

Our experiments use parallel Farsi-Spanish sentences extracted from the *OpenSubtitles2018*[1] collection (Tiedemann 2012) containing 1M sentences for training as well as 10K and 20K sentences for validation and test steps, respectively. The Spanish word embeddings (for the target language) are trained employing *Wikipedia* dumps[2] with *Gensim*[3]. We utilize *word2vec* (Mikolov et al. 2013) as a training method to obtain word embeddings. We use OpenNMT-py[4] model (Kelin et al. 2017) on top of PyTorch, based on a bi-directional 2-layer LSTM encoder-decoder with attention (Bahdanau, Cho, and Bengio 2015) (the decoders use global dot attention to the source vectors).

Training uses a batch size of 64 and Stochastic Gradient Descent (SGD) (Robbins and Monro 1951) with an initial learning rate of 0.01. We set the size of word embeddings as well as hidden layers to 500. We also set dropout to 0.1. We use a maximum sentence length of 50 and shuffle mini-batches as we proceed.

Similar to prior work (Jean et al. 2015), we select the best parameters with the smallest validation cost. Also, we employ BLEU (Papineni et al. 2001) as well as METEOR (Denkowski and Lavie 2014) as our evaluation metrics.

In the first set of experiments, we investigated the effects of different training strategies using cross-entropy as well as word embeddings cost functions. The goal is to determine the best practice among these strategies in attentional NMT training. We compared the baseline cost function (C_{ent}) to the word embeddings cost function (C_{emb})[5], as well as a combination of these functions (C_{com}). We select one specific cost (e.g., cross-entropy cost and embedding cost):

$$C_{com} = C_{ent} + C_{emb} \tag{14}$$

We add some trainable linear weights for the cost:

$$C_{com} = \alpha\, C_{ent} + (1 - \alpha)\, C_{emb} \tag{15}$$

We also investigated pre-training[6] with C_{ent} followed by training with C_{emb}. Using C_{emb} alone (without pre-training) does not work well, as the training process is likely to be trapped in a weak local minimum. We set the size of the target vocabulary to 20K. Since the performance of NMT systems often suffers in low-resource scenarios where sufficiently large-scale parallel corpora cannot be obtained, pre-trained word embeddings have proven to be invaluable for improving performance in natural language analysis tasks,

[1] http://opus.nlpl.eu/OpenSubtitles-v2018.php

[2] https://dumps.wikipedia.org

[3] https://radimrehurek.com/gensim

[4] https://github.com/OpenNMT/OpenNMT-py

[5] C_{emb} is subject to local minimum issues, which is a standard problem in NMT.

[6] Pre-trained embeddings used for distance calculation have a larger vocabulary so they can be used for OOV words.

Vocab	Cost	Pre-train	BLEU	METEOR
20,000	C_{ent}	None	28.88	35.65
20,000	C_{com}	None	30.12	36.61
20,000	C_{com}	C_{ent}	30.10	36.44
20,000	C_{emb}	C_{ent}	31.24	36.56

Table 1: Translation results applying 20K target vocabulary.

Vocab	Cost	Pre-train	BLEU	METEOR
1,000	C_{ent}	None	15.29	19.71
1,000	C_{com}	None	15.55	20.05
1,000	C_{com}	C_{ent}	15.79	19.93
1,000	C_{emb}	C_{ent}	15.84	21.45

Table 2: Translation results applying 1K target vocabulary.

Vocab	Cost	Pre-train	BLEU	METEOR
20,000	C_{ent}	None	31.97	55.21
20,000	C_{com}	None	32.12	55.96
20,000	C_{com}	C_{ent}	34.56	57.08
20,000	C_{emb}	C_{ent}	34.82	57.11

Table 3: Translation results applying 20K target vocabulary.

Source (Farsi)	سعی شد که عنصر مورب محدود از تصویر مقطعی از چشم واقعی ساخته شود
Target (Spanish)	Se intentó la preparación de la malla de elementos finitos a partir de las imágenes de la sección transversal del globo ocular real
C_{ent} output	Se intentó que la malla de elementos finitos se hiciera a partir de la imágen de la sección transversal de <UNK> real
C_{emb} output	Se intentó que la malla de elementos finitos se hiciera a partir de la imágen de la sección transversal del ojo real

Figure 1: Farsi-Spanish translation example 1.

Source (Farsi)	الگوی جریان در مخزن و شکل قالب ها
Target (Spanish)	El patrón del flujo en el depósito y la forma de las molduras
C_{ent} output	Patrónes de flujo en reservorio y <UNK> formas
C_{emb} output	El patrónes de flujo en el depósito y la forma de extensión

Figure 2: Farsi-Spanish translation example 2.

which often suffer from paucity of data. However, their utility for NMT has not been extensively explored. We examine pre-training with the baseline cost followed by training with our proposed cost. Using C_{emb} without pre-training does not work due to the aforementioned local minimum.

In the second set of experiments, we test a small target vocabulary to investigate the robustness in a small-size vocabulary condition. We set the size of the target vocabulary to 1K. Tables 1 and 2 show the results:

As seen in Tables 1 and 2, all methods using C_{emb} resulted in higher BLEU and METEOR scores than the baseline using only C_{ent}. These results confirm that the word embeddings cost function is effective with a relatively small target vocabulary (small-size vocabulary condition). The method that uses only C_{emb} after baseline pre-training showed significant improvements of +1.74 points in METEOR. These results suggest that the C_{emb} method works well with a limited vocabulary condition.

We also investigated the results in another language pair (as a high-resource scenario) to examine whether the advantage of our new cost function (C_emb) depends on a specific language. So, we conducted the same experiments using the WMT'18[7] Spanish-English dataset with the target language vocabulary size of 20K. Table 3 shows the results for Spanish-to-English which can be compared with the Farsi-Spanish results shown in Tables 1 and 2.

The results are similar but with a greater improvement due to the use of C_{emb} with pre-training. As experiment with Farsi-Spanish parallel corpus, all methods using C_{emb} improve translation accuracy on BLEU and METEOR metrics, especially using only C_{emb} after C_{ent} pre-training. The BLEU gains for Spanish-to-English translation are larger than those for Farsi-to-Spanish. This result suggests that the C_{emb} method is beneficial for not just one language pair.

[7]https://www.statmt.org/wmt18/translation-task.html

Discussion

We encode words using BPE with 32K merge operations to achieve an open vocabulary. OpenNMT-py enables substitution of OOV words with target words that have the highest attention weight according to their source words. When words are not found, a copy mechanism copies source words to the position of the not-found target word. We note that OpenNMT-py is selected over Transformer because it enables substitution of OOV words with target words that have the highest attention weight according to their source words.

The experimental results demonstrate the advantage of employing the word embeddings cost function for Farsi-Spanish low-resource NMT, most notably in the generation of similar words due to relaxed constraints in the cost function. Figures 1 and 2 illustrate two translation examples in a Farsi-to-Spanish experiment with a vocabulary size of 20K. These examples are generated by utilizing the baseline model (C_{ent}) as well as word embeddings model, which use C_{emb} after C_{ent} pre-training.

Figure 1 demonstrates a case where the target sentence includes the word *ocular*, but this is replaced with the special token <UNK> as an OOV word. In this case, the baseline translation result contains an OOV word that corresponds to a word that means *ocular*. By contrast, the model trained using our new cost function generated *ojo* instead of <UNK>. This suggests that our method enables generation of a reasonable word choice for low-frequency words. In Figure 2, the target sentence similarly includes a <UNK> token

corresponding to a word that means *molduras*. The C_{emb} model outputs a paraphrase *la forma de extensión* utilizing limited in-vocabulary words for the phrase *la forma de las molduras*.

Related Work

Prior work (Elbayad, Besacier, and Verbeek 2018) uses word vectors for smoothing a cost function in neural network-based language modeling. Their method aims at optimizing the conditional log-probability of improved output sentences sampled from the reward distribution. The rewards are defined based on the cosine similarity in a semantic word embedding space. They improve the results on image captioning and MT with both token-level and sequence-level rewards. They find that sequence-level rewards yield better performance improvement on MT tasks.

An alternative approach (Chousa, Sudoh, and Nakamora 2018) aims to minimize a weighted sum of distances from a reference word in a vector space, considering all the other words instead of some sampled words. Their token-level cost significantly improves upon the results above (Elbayad, Besacier, and Verbeek 2018), but the token-level rewards bring smaller improvement on MT tasks.

Another method (Sennrich, Haddow, and Birch 2016) reduces OOV words through the effective use of subwords and demonstrates that a subword-based system achieves higher performance than a word-based system for translating rare words. Their softmax cross-entropy still yields a generation probability that approaches 1 for the correct word and 0 for all other words regardless of their meaning. This approach also does not tackle our problem directly.

Various approaches have been proposed for word embeddings; *FastText* (Bojanowski et al. 2017), *word2vec* (Mikolov et al. 2013) and *Glove* (Pennington, Socher, and Manning 2014). Several of these yield vectors of syntactically and semantically similar words that are close to each other. Since embeddings are a key tool in transfer learning, techniques such as *ELMo* (Peters et al. 2018) have been proposed for embedding words in real vector space using LSTMs. Such approaches are trained on a language modeling objective and have beaten previous performance benchmarks, with a potential 10x reduction in training data.

BERT (Devlin et al. 2019) is proposed as an alternative to ELMo, targeting a different training objective: "masked language modeling." The motivation behind this work is the inability of prior approaches to take into account both left and right contexts of the target word due to left-to-right operation associated with the language modeling objective. For example, ELMo, simply concatenates the left-to-right and right-to-left information, which means the representation cannot exploit both contexts simultaneously. Our idea is complementary and would potentially assist systems like BERT and ELMo with OOVs in a low-resource context. However, since ELMo and BERT are context dependent, doing this requires the model that was used to train the vectors even after training, since the models generate the vectors for a word based on context.

Our work differs from those described above in that we adopt a new cost function for low-resource NMT, defined as a weighted average of distances between word vectors of a reference word and the others in the target language vocabulary. The weights are assigned by the generation probabilities in the softmax layer, yielding a small penalty for similar words and a large penalty to dissimilar words.

Label smoothing (Pereyra et al. 2017), as a solution to alleviate hard-target problems, is limited in that labels are smoothed with a uniform distribution; this is overcome by our model's distribution and incorporation of information about ratios between incorrect classes. Scheduled sampling (Bengio et al. 2015), as an alternative solution, is limited by *argmax* discontinuity and is thus unable to penalize errors made in previous steps. Our approach overcomes this by enabling fast and stable training, while also overcoming alignment limitations to reduce noise in the training signal. Cost definitions (Elbayad, Besacier, and Verbeek 2018), yet another solution, generally optimizes conditional log-probability of augmented output sentences sampled from the reward distribution, thus yielding only small improvements in token-level rewards in MT tasks. Our approach yields greater improvements by minimizing a weighted sum of distances from a reference word in vector space and considering all other words instead of a subset of sampled words.

Conclusions and Future Work

We applied a new cost function for Farsi-Spanish low-resource attention-based NMT using a weighted average of distance between a reference word and all the other target words in semantic space. Experimental results demonstrate advantages of our new method for computing translation accuracy, and for robust word selection considering semantic similarity in a limited-vocabulary condition. Experiments further demonstrate the advantage of using the new cost function in NMT, especially in generating similar words with the help of relaxed constraints in the loss function.

Future steps for this work include: (1) calculation of an efficient cost function over target language words; (2) use of different types of word embeddings; and (3) additional in-depth evaluation of attentional NMT through human-in-the-loop and task-oriented evaluations.

Acknowledgments

We would like to acknowledge the financial support received from the School of Science and Engineering, Tulane University of Louisiana (USA).

References

Ahmadnia, B.; Kordjamshidi, P.; and Haffari, G. 2018. Neural machine translation advised by statistical machine translation: The case of farsi-spanish bilingually low-resource scenario. In *Proceedings of the 17th IEEE International Conference on Machine Learning and Applications*, 1209–1213.

Ahmadnia, B.; Serrano, J.; and Haffari, G. 2017. Persian-spanish low-resource statistical machine translation through english as pivot language. In *Proceedings of the 9th International Conference of Recent Advances in Natural Language Processing*, 24–30.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of Advances in Neural Information Processing Systems*, 1171–1179.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Cho, K.; Van merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, 1724–1734.

Chousa, K.; Sudoh, K.; and Nakamora, S. 2018. Training neural machine translation using word embedding-based loss. *CoRR* abs/1807.11219.

Denkowski, M., and Lavie, A. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 376–380.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings in North American chapter Association for Computational Linguistics - Human Language Technology*, 4171–4186.

Dorr, B. J.; Pearl, L.; Hwa, R.; and Habash, N. 2002. Duster: A method for unraveling cross-language divergences for statistical word-level alignment. In *Proceedings of the 5th conference of the Association for Machine Translation in the Americas*.

Dorr, B. J. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics* 20(4):597–633.

Elbayad, M.; Besacier, L.; and Verbeek, J. 2018. Token-level and sequence-level loss smoothing for rnn language models. In *Proceedings of the the 56th Annual Meeting of the Association for Computational Linguistics*, 2094–2103.

Gehring, J.; Auli, M.; Grangier, D.; and Dauphin, Y. 2017. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 123–135.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Jean, S.; Cho, K.; Memisevic, R.; and Bengio, Y. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, 1–10.

Kelin, G.; Kim, Y.; Deng, Y.; Senellart, J.; and Rush, A. M. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics*, 67–72.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1412–1421.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations Workshop*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2001. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1532–1543.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, L.; and Hinton, G. 2017. Regularizing neural networks by penalizing confident output distributions. In *Proceedings of the 5th International Conference on Learning Representations*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proceedings in North American chapter Association for Computational Linguistics - Human Language Technology*, 2227–2237.

Robbins, H., and Monro, S. 1951. A stochastic approximation method. *Annals of Mathematical Statistics* 22:400–407.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 1715–1725.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Proceedings of Advances in Neural Information Processing Systems*, 3104–3112.

Tiedemann, J. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, 5998–6008.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv* abs/1609.08144.