Sentence Weighting for Neural Machine Translation Domain Adaptation

Shiqi Zhang **Deyi Xiong***
School of Computer Science and Technology, Soochow University, Suzhou, China
zzssq77@163.com, dyxiong@suda.edu.cn

Abstract

In this paper, we propose a new sentence weighting method for the domain adaptation of neural machine translation. We introduce a domain similarity metric to evaluate the relevance between a sentence and an available entire domain dataset. The similarity of each sentence to the target domain is calculated with various methods. The computed similarity is then integrated into the training objective to weight sentences. The adaptation results on both IWSLT Chinese-English TED task and a task with only synthetic training parallel data show that our sentence weighting method is able to achieve an significant improvement over strong baselines.

1 Introduction

Neural machine translation (NMT) has achieved more satisfactory performance than statistical machine translation (SMT) on many language pairs with various advantages over SMT, such as no pipeline-style training, more fluent translations and so on. However, it is still confronted with a big challenge in domain adaptation as the translation quality of NMT is heavily dependent on the quantity of training data and the relevance between the training data and the in-domain testing data. The training corpus usually varies across domains, where out-of-domain instances that are relevant to the target domain are beneficial for training while those which are irrelevant to the in-domain may deteriorate translation quality. The widely-used domain adaptation method in NMT is to fine-tune an existing out-of-domain model with the in-domain data (Luong and Manning, 2015). However, the fine-tuning method is of no consideration for the harm caused by irrelevant out-of-domain data. It also tends to overfit rapidly due to the small size of the in-domain data.

In this paper, we propose a sentence weighting method that evaluates the weights of sentences with respect to the relevance of sentences to the target domain. We train NMT models by assigning each sentence with a corresponding weight computed according to a domain similarity metric. Domain similarity has been successfully used in some tasks such as parsing, knowledge adaptation, etc.(Plank and Van Noord, 2011; Ruder et al., 2017). We employ domain similarity to measure relevance between the sentences of out-of-domain and in-domain data. In this way, we take into consideration both the goodness and the badness brought by out-of-domain data, in a weighting fashion. Additionally, different from current sentence weighting methods used for NMT that score sentences by making use of other toolkits like the SRI Language Modeling Toolkit (Stolcke, 2002), or an RNN classifier (Chen et al., 2017), our method exploits the information from the NMT system itself. This means that we do not need to train extra toolkits. We also examine the effectiveness of the proposed sentence weighting method on NMT trained with only synthetic parallel data, which is beneficial for the low-resource domain translation. Our method can also be used in back translation and in conjunction with other training methods.

* Corresponding author

Proceedings of the 27th International Conference on Computational Linguistics, pages 3181–3190
Santa Fe, New Mexico, USA, August 20-26, 2018.

Experiments show clear gains on the IWSLT Chinese-English task. Comparing to the previous sentence weighting method (Wang et al., 2017b), we achieve the highest improvement of 1.71 BLEU among four test sets, and obtain an average gain of 1.42 BLEU over Wang et al. (2017b)'s method. The experiments on NMT trained with only synthetic parallel data further confirm the effectiveness of our sentence weighting method on domain adaptation.

The paper is organized as follows. Section 2 overviews related work. In section 3, we give a background introduction of NMT. We present our sentence weighting method in Section 4. In section 5, we introduce the experimental setting and the NMT model configurations. In section 6 we show and discuss the results of our two groups of experiments, followed by our conclusion and future works in section 7.

2 Related Work

A lot of investigations have already been conducted for domain adaptation in SMT while few in neural machine translation. These methods can be roughly categorized into two classes: the model-level and data-level method.

At the model level, combining multiple translation models in a weighted manner is used for SMT domain adaptation. For NMT, fine tuning, model stacking and muti-model ensemble have been explored (Sajjad et al., 2017). Luong and Manning (2015) propose a fine-tuning method, which continues to train the already trained out-of-domain system on the in-domain data. Model stacking is to build an NMT model in an online fashion, training the model from the most distant domain at the beginning, fine-tuning it on the closer domain and finalizing it by fine-tuning it on the in-domain data. Muti-model ensemble combines multiple models during decoding using a balanced or weighted averaging method.

At the data level, traditional domain adaptation approach can be done by data selection, data weighting or data joining. Data selection approaches select data similar to the in-domain data according to some criteria. Normally, the out-of-domain data can be scored by a model trained on the in-domain data and out-of-domain data. For example, a language model can be used for scoring sentences (Axelrod et al., 2011). Data weighting methods weight each item which can be a corpus, a sentence or a phrase, and then train SMT models on weighted items.

Although some existing SMT domain adaptation techniques can be directly applied to NMT, it is challenging for applying data weighting to NMT. For NMT, the data selection approach can also be used. Wang et al. (2017a) employ the data selection method for domain adaptation, which uses sentence embeddings to measure the similarity of a sentence pair to the in-domain data. A recent method to apply sentence weights to NMT is cost weighting (Wang et al., 2017b; Chen et al., 2017). The NMT objective function is updated by sentence weighting when computing the cost of each mini-batch during NMT training. Wang et al. (2017b) exploit an in-domain language model (Axelrod et al., 2011) to score sentences. Chen et al. (2017) use a classifier to assign weights for individual sentences pairs.

Domain control uses word-level domain features in the word embedding layer, aiming to allow a model to be built from a diverse set of training data to produce in-domain translations (Kobus et al., 2017).

3 NMT

In this paper, we use the vanilla attention-based NMT (Bahdanau et al., 2014), which we will briefly summarize here.It is built on a recurrent neural networks (RNN) in an encoder-decoder framework (Sutskever et al., 2014; Pascanu et al., 2013).

Given a source sentence $x = (x_1, x_2, ..., x_m)$ and its corresponding target sentence $y = (y_1, y_2, ..., y_n)$, NMT employs the encoder-decoder framework to jointly train to maximize the conditional probability $p(y|x)$. To solve the problem that a basic encoder-decoder framework deteriorates rapidly as the length of input sentence increases, the attention mechanism (Bahdanau et al., 2014; Luong et al., 2015) is proposed.

The encoder employes a bi-directional RNN to encode the source sentence x into a sequence of hidden states h. h is the concatenation of the forward hidden state $\overrightarrow{h_i}$ and backward hidden state $\overleftarrow{h_i}$. Each hidden state h_i is computed from the previous hidden state h_{i-1} and the current source word x_i as follows.

$$\overrightarrow{h_i} = f(\overrightarrow{h}_{i-1}, x_i)$$

The decoder reads the hidden states and predicts the target translation by maximizing the conditional log-probability of the correct translation y. Each word y_i is predicted based on a recurrent hidden state s_i and a context vector c_i that aims at capturing relevant source-side information. c_i is computed as a weighted sum of the annotation h_i,

$$c_j = \sum_{j=1}^{m} \alpha_{ji} h_i$$

The weight α_{ji} of each annotation h_i is a normolized output from a softmax operation with a two layer feed-forward neural network ϕ. The weight α_{ji} of each annotation h_i indicates the probability of the target y_j being aligned to the source x_i.

$$\alpha_{ji} = \frac{exp(\phi(s_{j-1}, h_i))}{\sum_{k=1}^{n} exp(\phi(s_{j-1}, h_k))}$$

The probability of the target sentence y given a source sentence x is modeled as

$$P(y|x) = \prod_{j=1}^{m} P(y_j|s_j, y_{j-1}, c_j)$$

with s_j being the decoder state, c_j being the context, y_{j-1} being the last generated word.

Given a parallel corpus D, the parameters θ are trained to maximize the conditional probabilities of all sentences:

$$J = \sum_{(x,y)\in D} log(y|x)$$

4 Sentence Weighting for NMT

We evaluate domain adaptation methods on two different kinds of scenarios. One is for the usual domain adaptation that adapts a model trained on a rich-resource out-of-domain corpus to the in-domain with limited data. The other is for low-resource domain translation task, where only synthetic parallel data are available. For the latter, all the data are from the same domain while there are some differences between the true data and pseudo data. There also must be some mismatch between pseudo data and test data. For this kind of translation task, we can consider the development set as in-domain data, and the whole training data as out-of-domain data, though actually all data are from the same domain.

4.1 Sentence Weighting for Standard Domain Adaptation

We evaluate the out-of-domain sentences according to their domain similarity to the in-domain data. We consider the distribution of sentences in one domain corpus as the representation of the domain. Our hypothesis is that the average of all sentence embeddings (Wang et al., 2017a) in the domain space represents the core of the domain. In this way, we define the similarity measure between a sentence and a domain into that of the sentence to the core of the domain. The sentence embedding (Wang et al., 2017a) s_i is the representation of the initial hidden state for the decoder (Bahdanau et al., 2014). We can measure the Euclidean distance between the sentences and the domain with a method used by Wang et al. (2017a). In addition, we propose a new sentence weighting method using Jensen-Shannon (JS) Divergence (Lin, 1991) to measure the domain similarity.

We use softmax function to transform sentence embeddings into a probability distribution. Given a domain corpus of size N where each sentence embedding s_i is the initial hidden state for the decoder which is transformed from the embedded source sentence vector (Wang et al., 2017a), the probability of a sentence is computed as follows.

$$\sigma_{s_i} = \frac{exp(s_i)}{\sum_{k=1}^{k=N} exp(s_k)}$$

We therefore choose the Jensen-Shannon (JS) Divergence as the similarity function to measure the similarity between a sentence and a domain based on the distribution..

The Jensen-Shannon Divergence is a probabilistically-motivated function. It is symmetric and computes the Kullback–Leibler (KL) divergence between q and r, and their average. We use the JS divergence defined in (Lee, 2001):

$$JS(q,r) = \frac{1}{2}\left[D(q||avg(q,r)) + D(r||avg(q,r))\right]$$

where $D(q||r)$ is the KL divergence, which is a classical measure of "distance" between two probability distributions, and is defined as:

$$D(q||r) = \sum_{y} q(y) log \frac{q(y)}{r(y)}$$

We use JS Divergence to rank the out-of-domain sentences as follows:

$$-(JS_I(\sigma_{s_i}) - JS_O(\sigma_{s_i}))$$

where $JS_I(\sigma_{s_i})$ is the JS divergence between sentence representation σ_{s_i} and the in-domain representation. $JS_O(\sigma_{s_i})$ is the JS divergence to the out-of-domain data.

In practice, we use bilingual JS Divergence to compute the similarity metric, partially inspired by (Axelrod et al., 2011; Wang et al., 2017a; Wang et al., 2017b). Bilingual JS Divergence indicates that we takes into account both similarities on the source and target side of the corpus, which is calculated as follows.

$$\begin{aligned}\alpha_i = JS_{Osrc}(\sigma_{s_i}) - JS_{Isrc}(\sigma_{s_i}) \\ + JS_{Otrg}(\sigma_{s_i}) - JS_{Itrg}(\sigma_{s_i})\end{aligned}$$

$w(s_i)$ is a normalized output from a Min-Max Normalization (Priddy and Keller, 2005) over α_i. $w(s_i)$ estimates the weight of sentence s_i, in range of [0,1].

$$w(s_i) = \frac{\alpha_i - min(\{\alpha_i\}_{i=1}^{N})}{max(\{\alpha_i\}_{i=1}^{N}) - min(\{\alpha_i\}_{i=1}^{N})}$$

To train NMT with the weighted sentences, we update the objective function as follows:

$$J^* = \sum_{(x,y)\in D} w(s) log(y|x)$$

4.2 Sentence Weighting for NMT Trained with Only Synthetic Parallel Data

In domain adaptation, we measure the similarity between sentences in out-of-domain and the in-domain. We also test our sentence weighting method on synthetic parallel data. The existing pseudo parallel corpora is constructed with true and synthetic sentence pairs. The pseudo

IWSLT ZH-EN	#sentences
TED training (in-domain)	210k
LDC training	1.25M
TED dev 2010	0.9k
TED tst 2010	1.5k
TED tst 2011	1.4k
TED tst 2012	1.7k
TED tst 2014	1.3k

Table 1: Statistcs of the data used for the IWSLT task

parallel corpora can be source-originated, target-originated or mixture of them. In the low-resource domain translation task , we evaluate the similarity between sentences in training data and the development set. We use the bilingual JS Divergence method as follows:

$$\alpha_i = -JS_{Isrc}(\sigma_{s_i}) - JS_{Itrg}(\sigma_{s_i})$$

We regard all training data in the low-resource domain as out-of-domain data. To improve the translation performance, we apply a different cost weighting method by adding 1 to the normalized probability (Chen et al., 2017). This is to give the original training data a bonus to some degree. We use the following objective function instead:

$$J^* = \sum_{(x,y)\in D} (1 + w(s))log(y|x)$$

5 Experiments

The proposed methods were evaluated on two scenarios: standard domain adaptation and translation with synthetic parallel data which can be cast as a domain adaptation task. We will detail the experiment settings and results in this section.

5.1 Domain Adaptation

5.1.1 Data

For the Chinese-to-English translation domain adaptation task,we used an LDC news corpus as the out-of-domain data for the IWSLT 2017 (mainly contains TED talks) target domain corpus. The LDC bilingual corpus contains 1.25M sentence pairs extracted from LDC corpora, with 27.9M Chinese words and 34.5M English words. The IWSLT 2017 in-domain corpus contains 210k and 0.9k sentence pairs, for training and development set, respectively. We chose the TED TST2010, TED TST2011, TED TST2012, TED TST2014 datasets as our test sets. Statistics on the data of this task are shown in Table 1.

5.1.2 Setting

We used the case-insensitive 4-gram NIST BLEU score as our evaluation metric (Papineni et al., 2002) and the script 'mteval-v11b.pl' to compute BLEU scores. For the efficient training of the neural networks, we limited the source (Chinese) and target (English) vocabularies to the most frequent 30k words, covering approximately 97.7% and 99.3% words of the two corpora respectively. All the out-of-vocabulary words were replaced with a special token UNK. The dimension of word embedding was 620 and the size of the hidden layer was 1000. All other settings were the same as in (Bahdanau et al., 2014). The maximum length of sentences that we used to train the NMT model in our experiments was set to 50, for both the Chinese and English sides. Additionally, during decoding, we used the beam-search algorithm and set the beam size to 10. The model parameters were selected according to the maximum BLEU points on the development set.

Methods	tst2010	tst2011	tst2012	tst2014	Avg
In+Out	17.26	18.73	17.43	16.1	17.38
Wang et al. (2017b)	17.1	19.35	17.26	16.69	17.6
JS	17.57	19.74	17.84	17.78	18.23(+0.63)
ED	18.81	20.45	18.81	18.02	19.02(+1.42)

Table 2: IWSLT Chinese-English results

5.1.3 Results

Experiment results are shown in Table 2. "In+Out" indicates that the NMT training corpus is the mixture of in-domain and out-of-domain data. We also compared Wang et al. (2017b)'s sentence weighting method (hereafter WM), which uses a language model to score sentences. "JS" and "ED" indicate the methods that we proposed in Section 4.1 with similarity measured by Jensen-Shannon Divergence and Euclidean distance functions, respectively.

Our "JS" method achieves improvements over NMT (In+Out) by 0.31 - 1.68 BLEU. It also outperforms Wang et al. (2017b)' s sentence weighting method by 0.39 - 1.09 BLEU, and achieves an average gain of 0.63 BLEU over WM. Our "ED" method is much better than NMT (In+Out) by 1.38 - 1.92 BLEU, outperforming Wang et al. (2017b)' s sentence weighting method by 1.1 - 1.71 BLEU, and achieves an average gain of 1.42 BLEU over WM.

5.2 NMT with Only Synthetic Parallel Data

5.2.1 Data and Setting

We crawled Chinese and English monolingual sentences from E-commerce websites, including JD.com, Suning.com, Ebay and Amazon.com. We translated the crawled Chinese sentences into English and English sentences into Chinese using Google Translate API. This process provided us with a pseudo parallel corpus where either the source or the target side is correct. The final corpus contains 720k sentence pairs after filtering. We used this crawled and machine-translated corpus as our bilingual training data for E-commerce experiments. we also used BPE to process the training data.

The test and development set are from Alibaba. The development set contains 500 sentences and the test set contains 1,500 sentences. All model parameters are the same as indicated in Section 5.1.2, except for using Adadelta optimizer.

5.2.2 Results

The baseline is a system trained with the entire pseudo parallel corpus. We also apply Wang et al. (2017b)' s method in our experiment. The "JS" method obtains 0.7 BLEU points over the baseline, and outperforms WM by 0.44 BLEU points. The "ED" method is also better than WM.

In addition, we introduce a weighting and stacking training method that is inspired by model stacking method as described in Section 2. We trained several models in an online fashion, in the order of the similarity between the training corpus and the development set. We employed our sentence weighting approach described in Section 4 to calculate weights for sentences, and order training sentences by their weights. We start training the model on the entire training data. Similar to the model stacking method, we continue to train the model with half data of the ordered training corpora in the next epoch. Then we continue the training process with the quarter data in the same way. This weighting and stacking method outperforms the baseline by 1.15 BLEU points on the test set.

Methods	dev	tst
Baseline	15.46	13.35
Wang et al. (2017b)	16.04	13.61
JS	15.17	14.05
ED	15.34	13.73
Weighting and Stacking	15.36	14.50

Table 3: E-commerce Chinese-English results

SRC	各位女同胞们，大家好！你们还好吗?
REF	Hello, TEDWomen, what's up.
In+Out	ladies and gentlemen , hello ! are you okay ?
Wang	ladies and gentlemen , everybody ! are you good ?
JS	you guys , everyone ! are you okay ?
SRC	如果有什么类似受压迫奥运会，我肯定能拿金牌。
REF	if there was an Oppression Olympics, I would win the gold medal.
In+Out	if anything like the <UNK> Olympics, I will have a gold medal.
Wang	if there's something like the <UNK> Olympics, I certainly have a gold medal.
JS	if there were any similar olympic games, I would certainly take the gold medal.
ED	if there was something like the <UNK> Olympics, I would have a gold medal.

Table 4: Examples from the test set. Header "SRC" denotes source sentences. Header "REF" denotes reference translations. "In+out" represents the translations generated by "in+out" method, "Wang" by the sentence weighting method proposed by Wang et al. (2017b).

6 Analysis

6.1 Analysis on Sentence Representations

Evidence of the efficacy of the proposed similarity measure method can be visualized in Figure 1. In this figure, each symbol indicates one sentence which is represented by the normalized sentence embedding output σ_{s_i} from a softmax operation. We sample the out-of-domain (the LDC corpus) source sentences into 3 groups. Each group contains 20 sentences whose weights are around 0, 0.5 and 0.9 respectively. We also randomly select 20 sentences from the in-domain data.

From this figure we can see that the in-domain sentences are around the center of the in-domain, so do the out-of domain sentences. The visualization confirms our hypothesis in that the average of sentence embeddings represents the core of the domain (Section 4.1). What's more, we can find that the weight generated by our domain similarity method is reasonable. Sentences assigned similar weights are grouped together and far apart from those with different weights. In addition, the more similar the representation of a sentence to the core of in-domain, the higher the weight assigned to it.

6.2 Analysis on Translation Examples

We show two translation examples in Table 4 to provide a deep look into how the proposed method help adapt the model to the in-domain data. In the first example in Table 4 , the reference sentence has an informally utterance, where the greeting "what's up" indicates that it is in an informal atmosphere. As depicted in the Table 4, the greeting "ladies and gentlemen" translated by "In+Out" and "Wang et al. (2017b)" is used on formal occasions. "you guys" in "JS" method is uesd in an informal occasion. The "JS" method is able to adapt the more official LDC style to the spoken language domain.

The source and reference sentences in the second example in Table 4 are expressions in sub-

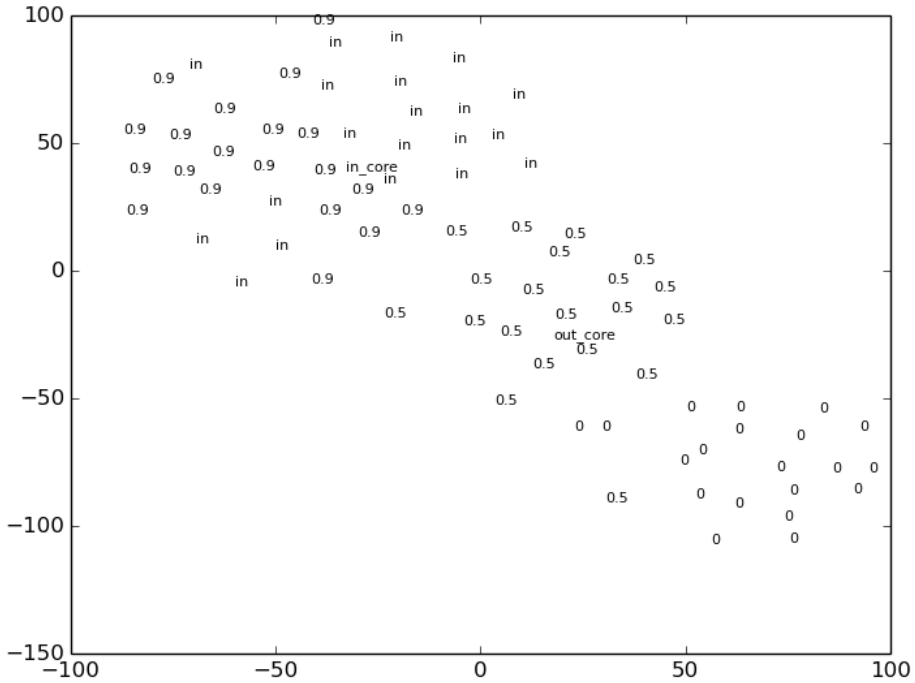

Figure 1: Visualization of sentence representations of both in-domain and out-of-domain examples by t-SNE. The symbols of "in", 0, 0.5, 0.9 in the figure represent the in-domain sentences, out-of-domain sentences with weights 0, 0.5 and 0.9, respectively. The symbol "in_core" denotes the in-domain center and "out_core" indicates the out-of-domain center.

junctive mood, which describes things that impossibly happen in real life. The translations of both "ED" and "JS" use the subjunctive mood and express the coincident meaning with the reference sentences. On the other hand, both "in+out" and "Wang et al. (2017b)" methods don't learn the right moods. As our LDC training data are in the news domain, sentences in this corpus seldom use the subjunctive mood. In the spoken language domain, the utterance is more free. Both the "ED" and "JS" methods have successfully adapted the model to the spoken language domain.

6.3 Analysis on the Terminology Adaptation

In domain adaptation, one of our goal is to extend generic NMT models to cover terminologies and styles in the target domain(Kobus et al., 2017). We analyse the distribution of terminologies on the IWSLT Chinese-English experiment. We calculate the proportion of in-domain and out-of-domain terminologies with respect to all words in the test set. We obtain terminology vocabularies according to differences of two sequences of words in both domains. The more in-domain terminologies and the fewer out-of-domain terminologies contained in the translations of the in-domain test sets, the better the domain adaptation is. The results are displayed in Table 5. The "ED" method with the best performance in terms of BLEU has successfully decreased the ratio of out-of-domain terminologies from 0.37% to 0.15% and increased the ratio of in-domain terminologies from 44.28% to 45.37%, indicating its ability to terminology adaptation.

7 Conclusions and Future Work

In this paper, we have proposed a sentence weighting method for neural machine translation on two different scenarios. We calculate the similarity of an out-of-domain sentence to the in-

Methods	ratio of O-D terminology(%)	ratio of I-D terminology(%)
In+Out	0.37	44.28
Wang et al. (2017b)	0.28	42.97
JS	0.20	43.80
ED	0.15	45.37

Table 5: The proportion of domain terminologies in the test sets of each method. "O-D" in header denotes out-of-domain. "I-D" represents in-domain.

domain with a variety of methods, including JS divergence and ED. The computed similarity scores are further incorporated into the training objective to weight sentences. The proposed method has obtained an achievement on both domain adaptation and NMT models trained with only synthetic parallel data where there is some mismatch between the training and test data. In the future, we would like to explore more accurate measures of domain similarity. We are also interested in the study of other weighting methods for NMT domain adaptation.

Acknowledgements

The present research was supported by the National Natural Science Foundation of China (Grant No. 61622209).We would like to thank three anonymous reviewers for their insightful comments and Alibaba for providing the test and development sets for our E-commerce experiments.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378. INCOMA Ltd.

Lillian Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *AISTATS*.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2013. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Barbara Plank and Gertjan Van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1566–1576. Association for Computational Linguistics.

Kevin L Priddy and Paul E Keller. 2005. *Artificial neural networks: an introduction*, volume 68. SPIE press.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2017. Knowledge adaptation: Teaching to adapt. *arXiv preprint arXiv:1702.02052*.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. *arXiv preprint arXiv:1708.08712*.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *In Proceedings of international conference on spoken language processing. Denver, Colorado, USA, September 2002*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017a. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 560–566.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017b. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488.