Word-based Domain Adaptation for Neural Machine Translation

Shen Yan, *Leonard Dahlmann*, *Pavel Petrushkov*, *Sanjika Hewavitharana*, *Shahram Khadivi*

eBay Inc.

{shenyan,fdahlmann,ppetrushkov,shewavitharana,skhadivi}@ebay.com

Abstract

In this paper, we empirically investigate applying word-level weights to adapt neural machine translation to e-commerce domains, where small e-commerce datasets and large out-of-domain datasets are available. In order to mine in-domain like words in the out-of-domain datasets, we compute word weights by using a domain-specific and a non-domain-specific language model followed by smoothing and binary quantization. The baseline model is trained on mixed in-domain and out-of-domain datasets. Experimental results on En $\rightarrow$ Zh e-commerce domain translation show that compared to continuing training without word weights, it improves MT quality by up to 3.11% BLEU absolute and 1.59% TER. We have also trained models using fine-tuning on the in-domain data. Pre-training a model with word weights improves fine-tuning up to 1.24% BLEU absolute and 1.64% TER, respectively.

1. Introduction

Domain adaptation (DA) techniques in machine translation (MT) have been widely studied. For statistical machine translation (SMT), several DA methods have been proposed to overcome the lack of domain-specific data. For example, self-training [1, 2] uses a MT system trained on general corpus to translate in-domain monolingual data as additional training sentences. Topic-based DA [3, 4] employs topic-based translation models to adapt for different scenarios. Data selection approaches [5, 6, 7, 8] first score the out-of-domain data using language model trained on both domain-specific and non-domain-specific monolingual corpora, then rank and select the out-of-domain data that are similar to in-domain data. Instance weighting methods [9, 10] score each sentence/domain using statistical rules, then train the MT models by giving sentence/domain-level scores.

Neural machine translation (NMT) has become state-of-the-art in recent years [11, 12, 13, 14, 15]. There are several research works on NMT domain adaptation. For example, back-translation methods [16] use a NMT model trained on the reverse direction to translate domain-specific monolingual data as additional training sentences. Fast DA approaches [13, 17] train a base model using mixed in-domain and out-of-domain datasets, then fine-tuning on in-domain datasets. Mixed fine-tuning [18] combines fine-tuning and multi-domain NMT. Similar to instance weighting in SMT, sentence/domain weighting methods [19, 20] can also be used for NMT domain adaptation based on different objectives. DA with meta information [21] is proposed to train topic-aware models using domain-specific tags for the decoder. Chunk weighting method [22] describes a way of selecting and integrating positive partial feedback from model-generated sentences into NMT training.

In this paper, we propose word-level weighting for NMT domain adaptation. We compute the word weights in out-of-domain datasets based on the logarithm difference of probability according to a domain-specific language model and non-domain-specific language model followed by smoothing and binary quantization. This gives the in-domain words in out-of-domain sentences higher weights and biases the NMT model to generate more in-domain-like words. Thus, the work presented in this paper can be viewed as a generalization of instance weighting. To remove noise in the word weights, we study the effectiveness of using smoothing methods. Specifically, a weighted moving average filter is proposed to apply smoothing to the computed word scores with its nearby words.

Experiments on En $\rightarrow$ Zh e-commerce domain translations tasks show that: 1) Domain adapted model with smoothed word weights gains significant improvement over non-smoothed weights; 2) Continuing training the model with computed word weights improves translation results significantly compared to continuing training without word weights; and 3) Compared to directly fine-tuning on in-domain datasets, fine-tuning after pre-training with word weights results in translation performance improvement on the in-domain e-commerce test set.

The rest of the paper is structured as follows. The approach and model we use is described in Section 2, where we first recap the NMT objective and then present the details of the proposed word-level weighting approach. Experimental results and discussions are presented in Section 3 and Section 4, followed by conclusions and outlook in Section 5.

2. Approach

We present word weighting objective on NMT before discussing how to generate the weights.

2.1. Objective

In this work we use attention-based neural machine translation model [11, 12, 14] for experiments. Given a parallel bilingual dataset *D*, the NMT model is trained to maximize the conditional likelihood *L* of a target sequence $y_1^T : y_1, \ldots, y_T$ given a source sequence $x_1^N : x_1, \ldots, x_N$:

$$L = \sum_{(x_1^N, y_1^T) \in D} \sum_{t=1}^{T} \log p(y_t | y_1^{t-1}, x_1^N) \tag{1}$$

Training objective (1) can be simply modified to word-level loss L_w with word weights w_t:

$$L_w = \sum_{(x_1^N, y_1^T, w_1^T) \in D} \sum_{t=1}^{T} w_t \log p(y_t | y_1^{t-1}, x_1^N) \tag{2}$$

The word weights w_t for a target sequence y_1^T can be 0 or 1. We set w_t = 1 for all in-domain sentences. For out-of-domain sentences, w_t = 1 means the word in the out-of-domain sentence is related to in-domain datasets (selected), w_t = 0 means it is not.

Our training objective (2) can be seen as a generalization of the original training objective (1) and instance weighting methods [19, 20]. The original loss (1) sets w_t = 1 for every word in all sentences. The instance-level loss can be expressed as giving a target sentence, $w_t = w \; \forall t$, where w is the weight for the sentence or the domain. Our training objective is similar to [22], however, instead of generating chunk-based user feedback for model predictions, we compute the word weights using language models trained on real target data.

2.2. Approaches to the objective

To compute discriminative word weights, we first follow the data selection methods in SMT [5]. To state this formally, let I be the domain-specific corpus, O be the non-domain-specific corpus, and y_t be the word in out-of-domain sentences at target position t. We denote by $P_I(y_t | y_{t-n}^{t-1})$ the per-word probability conditioned on previous $n-1$ words, according to a language model trained on I. Similarly, we denote by $P_O(y_t | y_{t-n}^{t-1})$ the per-word probability conditioned on previous $n-1$ words according to a language model trained on O. We can estimate $P_I(y_t | y_{t-n}^{t-1})$ and $P_O(y_t | y_{t-n}^{t-1})$ by training language models on I and O, separately. Therefore, the word scores s_t can be computed in the log domain:

$$s_t = \log P_I(y_t | y_{t-n}^{t-1}) - \log P_O(y_t | y_{t-n}^{t-1}) \tag{3}$$

Since the value of s_t is strongly correlated with the neighborhood words, it is worth investigating smoothing of the word scores before binary thresholding to remove the noise. Hence, a weighted moving average kernel:

$$\hat{s_t} = \sum_{k=\lfloor \frac{-L}{2} \rfloor}^{\lfloor \frac{L}{2} \rfloor} c_k s_{t+k} \tag{4}$$

is then applied to smooth word score s_t at each target position t. Here L is the kernel size and c_k are values of the kernel for $k \in [\frac{-L}{2}, \frac{L}{2}]$. In our experiments, we heuristically set the values of the kernel based on mean average with $c_k = c = \frac{1}{L}$ or gaussian distribution with $c_k = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-k^2}{2\sigma^2}}$, where we set σ to be the global variance of the word scores.

The special case of sentence-level weights can be expressed as $\hat{s_t} = \hat{s} \; \forall t$, where $\hat{s}$ is the averaged smoothed word scores for the target sentence y_1^T. In this case, the training objective (2) becomes equivalent to sentence weighting method from [20] with appropriately modified scoring function.

After smoothing the word scores, we finally binarize the smoothed word scores based on a threshold T:

$$w_t = \begin{cases} 1, & \text{if } \hat{s_t} \geq T \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

In our experiments we set the threshold $T = 0.5$ and only keep the words above the threshold. This means we select a word if $w_t = 1$ and do not select it if $w_t = 0$. Considering word weights w_t are gathered in a binary form during continuing training, the selected words would be good candidates that we want to extract from out-of-domain corpus O. In fact, word weights w_t are precomputed offline and used during the training. It can be set to any real value, depending on the way of thresholding.

2.3. Chunk-based weighting

Considering that the selected words in a target sentence might still be noisy and we select single random words, we alternatively experimented with selecting only the part (chunk) in the target sentence that has the longest consecutive weights (LCW) with $w_t = 1$. For each target sentence, we pick only one chunk and set all other weights to zero. See Figure 1 for an example. Then, because the surrounding context is also selected, the chunk is less likely to be noise. If there are multiple such chunks with the same length in the sentence, we simply randomly sample one of them. We found that the chunk-based approach in practice performs slightly better than word-level weighting.

3. Experiments

In this section, we conduct a series of experiments to study how well NMT performs when word-level weights are given for out-of-domain training data. We also study the effectiveness of the smoothing methods.

3.1. Datasets and data processing

We report the results on our in-house English-to-Chinese e-commerce item descriptions dataset. Item descriptions are

provided by private sellers and like any user-generated content, may contain ungrammatical sentences, spelling errors, and other type of noise. We first segmented the Chinese sentences with Stanford Chinese word segmentation tool [23] and tokenized English sentences with the scripts provided in Moses [24]. On both languages, we use subword units based on byte-pair encoding (BPE) [25] with 42,000 subword symbols learned separately for each language. For En-Zh we have 0.53M in-domain e-commerce sentence pairs and 5.15M sampled out-of-domain sentence pairs (UN, subtitles, TAUS data collections, etc.) that have significant n-gram overlap with the item description data. We validate our models on an in-house development set consisting of 3173 item descriptions, and evaluate on an in-house test set of 739 item descriptions using case-insensitive character-level BLEU [26] and TER [27] with in-house tools. For development and test sets, a single reference translation is used. Statistics of the data sets are reported in Table 1.

To compute our word weights we train a domain-specific 4-gram language model and a non-domain specific 4-gram language model using KenLM [28]. For the domain-specific language model, we collected domain-specific monolingual data from an e-commerce website, resulting in the number of 15M sentences. For the non-domain-specific language model, we use sampled LDC Chinese Gigaword (LDC Catalog No.: LDC2003T09) with 36M sentences. It should be noted that we train our language models on the word-level. In order to score a BPE-level corpus with such a language model, we score its words and copy this score for each of the subword units. After the word scores are computed, we then smooth them via a guassian distributed kernel with window size $L = 5$. We choose window size $L = 5$ considering that the language model is trained based on sequences of four words. We observed similar results with different window sizes, which is discussed in Section 4. Finally, we binarize the smoothed word scores into binary word weights by setting the threshold $T = 0.5$. The computed word weights are applied to the target side of out-of-domain sentences during the phase of continuing training. In order to get better translation results, we first trained the baseline model with mixed in-domain and out-of-domain data according to training objective 1, where no weights are used. We start our experiments by continuing training from this baseline model.

We implemented our NMT model using Tensorflow [29] library. The encoder is a bidirectional LSTM with size of 512 and the decoder is a LSTM with 2 layers of same size. All the weight parameters are initialized uniformly in $[-0.1, 0.1]$. We set dropout on RNN inputs with dropping probability 0.2. We train the networks with batch size 120 using SGD with initial learning rate 1.0 and gradually decaying to 0.1 after the initial 2 epochs.

3.2. Results

Statistics of the out-of-domain sentences/tokens selection after applying different types of weights are summarized in Table 2. Before the selection, the number of out-of-domain sentences is 5.15M and the number of tokens is 93.4M. When sentence-level weights are used, the sentences with $w_t = 0$ are ignored, resulting in the number of remaining sentences/tokens around 2.63M and 26.3M, respectively. When word-level weights are used, there are $1,279,927$ sentences where all word weights in the sentences are equal to zero. After removing these sentences, around 3.87M sentences are preserved and the number of selected tokens with word weights w_t=1 is around 36.6M. Given computed word weights, we alternatively choose only the chunk with the longest consecutive weights (LCW) where $w_t = 1$, resulting in chunk-level weights with the selected number of tokens further reduced to 25.8M.

We train a baseline NMT model on mixed in-domain and out-of-domain data with objective defined as Eq. 1 for 6 epochs. The data is mixed completely (mixed 0.53M in-domain e-commerce and 5.15M sampled out-of-domain sentence pairs) while training the baseline model. The baseline model initialized by a mix of in-domain/out-of-domain data can be regarded as a kind of "warm start". We have also tried training a baseline with out-of-domain data only and observed slightly worse result after fine-tuning on in-domain data (0.5 BLEU). Hence, we use the baseline model trained on a mix of in-domain/out-of-domain data in the following experiments. Given the baseline model, we then directly fine-tune on in-domain data for another 10 epochs or first continue training on the mixed data with sentence/chunk/word weights for 3 epochs and then fine-tune on in-domain data for 10 epochs. The model is saved after each epoch. We take the model which gives the best result on our development set for evaluation. Note that we always set word weights $w_t = 1$ for our in-domain dataset.

In Table 3, we show the effect of different types of weights on translation performance. First, the baseline trained on mixed in-domain and out-of-domain datasets gives 24.37% BLEU and 61.66% TER, respectively. Directly fine-tuning on in-domain dataset already improves the model due to the bias of the model towards in-domain data.

Continuing training on mixed datasets with previous objective defined in Eq. 1 shows insignificant changes in terms of BLEU and TER. However, introducing sentence-level weights improves the model from 24.37% to 25.79% BLEU and 61.66% to 60.82% TER, respectively. Compared to continuing training without weights, sentence-level weights are generated as described in Section 2.2, where w_t $\forall t$ are set to the same sentence weight $w \in \{0, 1\}$. We set the threshold equal to 0.5 and keep the sentences with weights above the threshold. The result from sentence-level feedback suggests that mining good out-of-domain sentences which are similar to in-domain datasets and dissimilar to out-of-domain datasets benefits model translation towards in-domain-like sentences even without fine-tuning on in-domain datasets.

The use of word-level weights improves the baseline model even better, from 24.37% to 26.14% BLEU and

Data set		e-commerce + out-of-domain	
Language		English	Chinese
Training	Sentences	5,689,989	
	Running words	97,266,344	96,480,106
	BPE vocabulary	33,484	45,867
Dev	Sentences	3173 (item descriptions)	
	Running words	51,130	48,900
Test	Sentences	739 (item descriptions)	
	Running words	19,034	18,262

Table 1: Corpus statistics for the e-commerce English→Chinese MT tasks.

Corpus	Sent. count	Token count
ood. sentences	5,153,191	93,427,867
+sent. weights	2,633,109	26,275,096
+word weights	3,873,264	36,617,395
+chunk weights	3,873,264	25,813,480

Table 2: Out-of-domain training corpus statistics. *ood. sentences* indicates the number of sentences/tokens in the out-of-domain corpus. *+sent. weights* indicates the number of selected out-of-domain sentences where the weights of the sentences are equal to 1. *+word weights* and *+chunk weights* indicate the statistics of selected out-of-domain sentences/tokens after applying word weights generation and LCW methods as described in Section 2.2 and 2.3.

61.66% to 60.34% TER, respectively. In this approach, the number of selected tokens is drastically reduced to 36.6M from 93.4M tokens, nearly 61% drop in number of tokens with improved translation performance. Word-level weights also outperform sentence-level weights by 0.35% in BLEU score and 0.48% in TER. It can be explained by the fact that each word in the sentences are given its own similarity to the in-domain datasets. Considering sentence-level weights set all words in a sentence with the same weight, even though part of the words in the sentences might not be related to the in-domain corpus, word-level weights are more accurate and effective.

Finally, chunk-level weights are generated from our word-level weights based on LCW. Here we aim to train the domain-adapted model from more consecutive segments rather than single selected words. On top of word-level weights, it improves by another 0.28% BLEU absolute and 0.24% TER, respectively. Out-of-domain sentences can be split into chunks which can be related to the in-domain and can be translated independently in terms of the context. The selection of the consecutive chunk with in-domain-like context can positively affect the training towards domain-adapted model. By focusing on in-domain related and out-of-domain unrelated part, word/chunk-level weights can effectively reduce the unnecessary noise in the out-of-domain training data. Compared to continuing training without word weights, we are able to further reduce the corpus by 72% tokens (25.8M vs. 93.4M selected tokens), resulting in an improvement of 2.11% BLEU absolute and 1.59% TER, respectively. It should also be noted that with similar number of tokens (25.8M vs. 26.3M), chunk-level weights outperforms sentence-level weights by 0.63% BLEU absolute and 0.72% TER.

Next, we further fine-tune the model with chunk-level weights and obtain further improvements of 0.88% BLEU absolute and 1.81% TER. Compared to directly fine-tuning on the baseline, continuing training the model with chunk-level weights and then fine-tuning improves translation results from 26.06% to 27.30% BLEU and 59.93% to 58.29% TER, respectively.

Results from the study on the effect of using different smoothing methods are shown in Table 4. The word weights generated without using smoothing methods, where $\hat{s}_t = s_t$, lead to poor translation quality of 21.38% from 24.37% BLEU and 66.25% from 61.66% TER, respectively. We need to smooth the word scores before thresholding because the values of $\log P_I(y_t|y_{t-n}^{t-1}) - \log P_O(y_t|y_{t-n}^{t-1})$ are noisy. If there are selected isolated words like ',' which have higher scores than the surrounding text, it may cause rare vocabulary problem after training.

The results from word weights computed from mean averaged filter and normal distributed filter are relatively close, 25.99% vs. 26.14% BLEU and 60.70% vs. 60.34% TER, respectively. These results are obtained via a filter with window size $L = 5$. In practice, we also tried setting window size $L = 3$ and $L = 7$, but didn't observe different results. We found that the surrounding word scores have to be considered for smoothing in order to make the word weights w_t less noisy as well as more precisely representing the similarity to the in-domain/out-of-domain.

Additionally, we also experimented with randomly selecting words in the out-of-domain sentences with binary mask. However, we observed a drop in the translation accuracy.

3.3. Examples

In Table 5, we show an example for which the system trained with word weights produces a better translation. The English sentence is "non-spill spout with patented valve". The

		Item descriptions	
No.	System description	BLEU [%]	TER [%]
1	Baseline	24.37	61.66
2	1 + continue training without word weights	24.31	61.69
3	1 + continue training with sentence weights	25.79	60.82
4	1 + continue training with word weights	26.14	60.34
5	1 + continue training with chunk weights	26.42	60.10
6	1 + fine-tuning on in-domain	26.06	59.93
7	5 + fine-tuning on in-domain	27.30	58.29

Table 3: E-commerce English → Chinese BLEU results on test set. *Baseline* is trained on mixed in-domain and out-of-domain data. *No. 2* is continuing training from baseline with objective defined as Eq. 1. *No. 3* is continuing training from baseline with sentence-level weights and *No. 4* is with word weights, as defined in Section 2.2. *No. 5* refers to assigning w_t using LCW method described in Section 2.3. *No. 6* is equivalent to directly fine-tuning on in-domain datasets starting from the baseline model and *No. 7* is equivalent to fine-tuning on in-domain datasets after *No. 5* is finished.

System	BLEU [%]	TER [%]
Baseline	24.37	61.66
+w.w. without smooth.	21.38	66.25
+w.w. (mean smooth.)	25.99	60.70
+w.w. (gauss. smooth.)	26.14	60.34

Table 4: Study on the effect of different smoothing methods for word weights generation. *Baseline* is the same as before. *w.w. without smoothing* means the word weights (w.w.) are computed without smoothing in the log domain. *w.w. (mean smooth.)* indicates smoothing the word scores via using a mean average filter before thresholding and *w.w. (gauss. smooth.)* indicates using a normal distributed filter before thresholding. The approaches regarding different smoothing methods are described in Section 2.2.

word "spout" is rare in our data, appearing in the out-of-domain training sentences only once. The Chinese side of this training example can be seen in Figure 1 together with the weights assigned to the individual words by our method. When smoothing is applied, isolated Chinese words such as "空气" ("air") are removed. With the longest consecutive words (LCW) method, the only remaining chunk is "防/溢出/喷口/内" ("inside the non-spills spout"), which is related to our in-domain data. The system with word weights is then trained only on this chunk on the target side, while the baseline model is trained on the entire sentence and generates inappropriate translations.

4. Discussions

The domain adaptation techniques (sentence-level/chunk-level/word-level) introduced in this paper are all derived from word weights generation. They aim to select out-of-domain sentences/chunks/words which are more related to in-domain corpus and unrelated to out-of-domain corpus. The word weights are computed prior to system tuning via the logarithm difference of LM probability scoring and are then used for tuning the sequence-to-sequence model. By measuring domain similarity with external criteria such as LM, this kind of out-of-domain data selection is able to highlight the in-domain-related and out-of-domain-unrelated parts and leads to less variation and errors in our e-commerce domain adaptation. In addition, the selected out-of-domain segments have to be smoothed in order to reduce noise.

5. Conclusions

In this work, we generate word-level weights by calculating the logarithm difference of the probability of two external language models for domain adaptation. This approach better selects the out-of-domain segments related to e-commerce domain, and requires fewer tokens for training. We experimented with continuing training models with sentence/chunk/word weights and show that they all give translation improvement in terms of BLEU and TER compared to continuing training without word weights. Experiments on our in-house English-Chinese datasets also show that continuing training with word weights then fine-tuning improves results over directly fine-tuning on baseline model.

In future, with the computed word weights as the initial parameters, we want to devise strategies to make online domain adaptation possible by dynamically updating word weights during training, which could in turn lead the in-domain data translation to better match its references.

6. References

[1] N. Ueffing and H. Ney, "Word-level confidence estimation for machine translation using phrase-based translation models," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, ser. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 763–770. [Online]. Available: https://doi.org/10.3115/1220575.1220671

System	Translation
Baseline	带专利阀的防溢出溅漏
+ word weights	带专利阀的防溢出喷口
Reference	带有专利阀门的防溢口

Table 5: Translations of ”non-spill spout with patented valve” produced by the baseline NMT system and the system trained with word weights. The last row shows the reference translation.

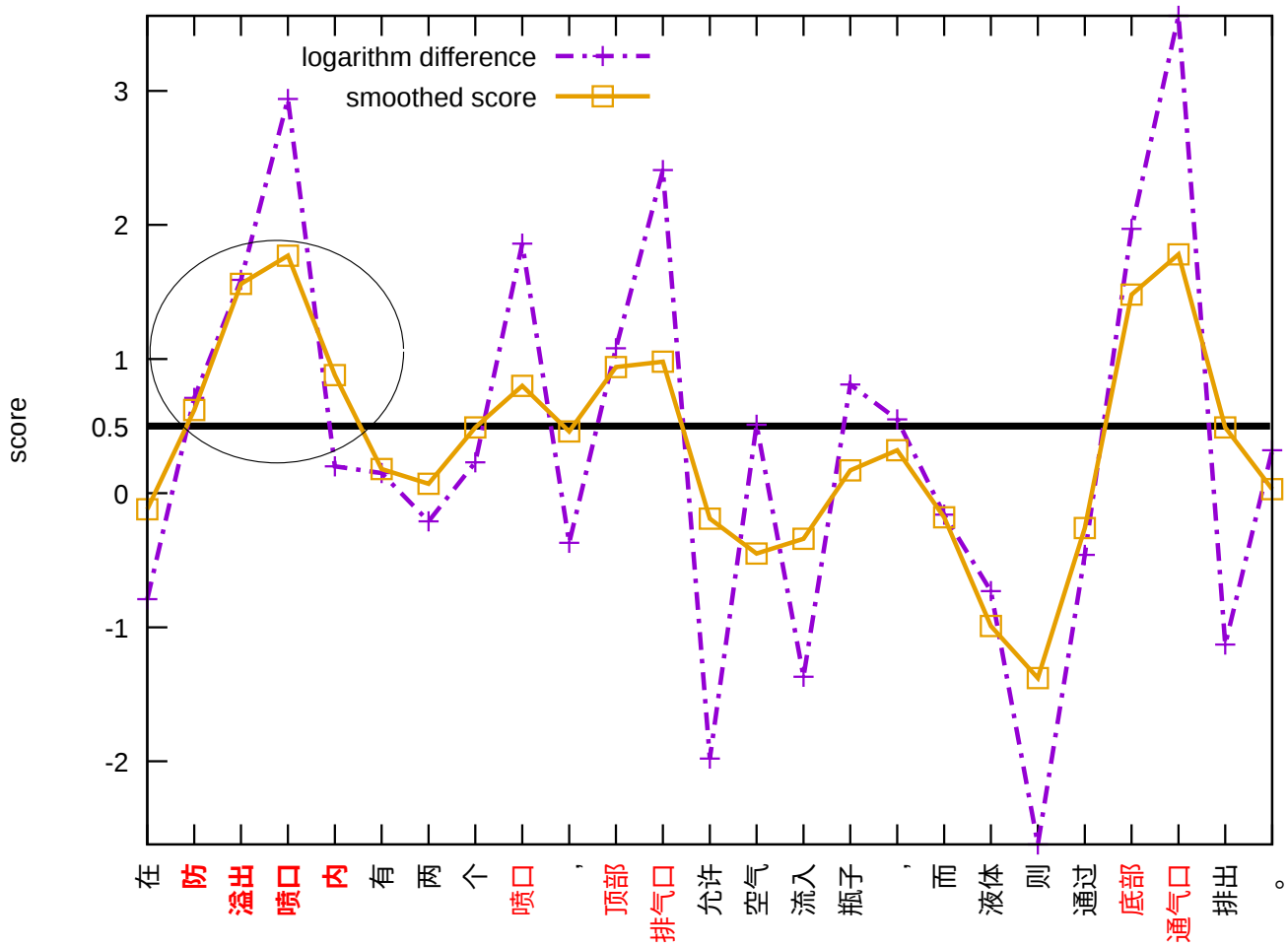

Figure 1: Our approach for word-based domain adaptation, with the target sentence on the bottom (the source sentence is ”Inside the non-spills spout there are two vents, where the top vent allows air to flow into the bottle while liquid smoothly pours out through the bottom vent.”). Above are displayed the word scores and the smoothed word scores. The black bold line indicates the threshold and the circle indicates which chunk is selected by LCW. After smoothing, the isolated random words are removed and the red words on the bottom are selected. The red bold words on the bottom are preserved after LCW.

[2] H. Schwenk, "Investigations on large-scale lightly-supervised training for statistical machine translation," in *IWSLT*, 2008.

[3] Y.-C. Tam, I. R. Lane, and T. Schultz, "Bilingual-lsa based lm adaptation for spoken language translation," in *ACL*, 2007.

[4] S. Hewavitharana, D. Mehay, S. Ananthakrishnan, and P. Natarajan, "Incremental topic-based translation model adaptation for conversational spoken language translation," in *ACL*, 2013.

[5] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*, ser. ACLShort '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 220–224. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858842.1858883

[6] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 355–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145474

[7] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *ACL*, 2013.

[8] N. D. H. S. S. J. A. A. S. Vogel, "Using joint models for domain adaptation in statistical machine translation," in *MT Summit*, 2015.

[9] S. Matsoukas, A.-V. I. Rosti, and B. Zhang, "Discriminative corpus weight estimation for machine translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, ser. EMNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 708–717. [Online]. Available: http://dl.acm.org/citation.cfm?id=1699571.1699605

[10] G. Foster, C. Goutte, and R. Kuhn, "Discriminative instance weighting for domain adaptation in statistical machine translation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 451–459. [Online]. Available: http://dl.acm.org/citation.cfm?id=1870658.1870702

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969173

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2015.

[13] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains," in *Proceedings of the International Workshop on Spoken Language Translation : December 3-4, 2015, Da Nang, Vietnam / Edited by Marcello Federico, Sebastian Stüker, Jan Niehues*. International Workshop on Spoken Language Translation, Da Nang (Vietnam), 3 Dec 2015 - 4 Dec 2015, Dec 2015.

[14] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[16] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *ACL*, 2016.

[17] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation," *CoRR*, vol. abs/1612.06897, 2016.

[18] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *ACL*, 2017.

[19] B. Chen, C. Cherry, G. Foster, and S. Larkin, "Cost weighting for neural machine translation domain adaptation," in *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, 2017, pp. 40–46. [Online]. Available: http://aclweb.org/anthology/W17-3205

[20] R. Wang, M. Utiyama, L. Liu, K. Chen, and E. Sumita, "Instance weighting for neural machine translation domain adaptation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 1482–1488. [Online]. Available: http://aclweb.org/anthology/D17-1155

[21] S. Khadivi, P. Wilken, L. Dahlmann, and E. Matusov, "Neural and statistical methods for leveraging meta-information in machine translation," in *MT Summit*, 2017.

[22] P. Petrushkov, S. Khadivi, and E. Matusov, "Learning from chunk-based feedback in neural machine translation," in *ACL*, 2018.

[23] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing chinese word segmentation for machine translation performance," in *Proceedings of the Third Workshop on Statistical Machine Translation*, ser. StatMT '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 224–232. [Online]. Available: http://dl.acm.org/citation.cfm?id=1626394.1626430

[24] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557769.1557821

[25] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *CoRR*, vol. abs/1508.07909, 2016.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135

[27] M. Snover, B. J. Dorr, R. F. Schwartz, and L. Micciulla, "A study of translation edit rate with targeted human annotation," 2006.

[28] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, ser. WMT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 187–197. [Online]. Available: http://dl.acm.org/citation.cfm?id=2132960.2132986

[29] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283. [Online]. Available: http://dl.acm.org/citation.cfm?id=3026877.3026899