INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Improving Isochronous Machine Translation with Target Factors and Auxiliary Counters

Proyag Pal[1,2]*, *Brian Thompson*[1], *Yogesh Virkar*[1], *Prashant Mathur*[1], *Alexandra Chronopoulou*[1,3]*, *Marcello Federico*[1]

[1]AWS AI Labs, USA
[2]School of Informatics, University of Edinburgh, Scotland
[3]Center for Information and Language Processing, LMU Munich, Germany

brianjt@amazon.com

Abstract

To translate speech for automatic dubbing, machine translation needs to be isochronous, i.e. translated speech needs to be aligned with the source in terms of speech durations. We introduce target factors in a transformer model to predict durations jointly with target language phoneme sequences. We also introduce auxiliary counters to help the decoder to keep track of the timing information while generating target phonemes. We show that our model improves translation quality and isochrony compared to previous work where the translation model is instead trained to predict interleaved sequences of phonemes and durations.

Index Terms: automatic dubbing, isochrony aware machine translation, target factors, auxiliary counters

1. Introduction

Automatic dubbing [1] aims to translate speech from video content (such as movies and TV shows) into a target language while maintaining isochrony, i.e. matching the speech and pause structure of the source speech in order to preserve time synchronization in the dubbed video. In the standard automatic dubbing pipeline, an automatic speech recognition (ASR) system transcribes the source audio into source language text, the text is translated into the target language by a machine translation (MT) system, after which a prosodic alignment (PA) module inserts pauses to segment the translated text, before a text-to-speech (TTS) system generates target language speech.

One drawback of this pipeline is the fact that since the machine translation system is unaware of isochrony constraints, it can generate translations which do not fit the timing of the source audio. After segmenting target text through the PA module, to ensure the segments fit the speech timing, the speaking rate has to be adjusted for the TTS system, often resulting in unnatural output speech.

Our goal is to jointly optimize translation quality and isochrony, i.e. predict translations and target-side timing information using the same model to generate translations of high quality while matching the source's speech timing. We achieve this using target factors [2], where alongside predicting phoneme sequences as the target, we also predict durations for each phoneme as a target factor. Additionally, we design auxiliary counters[1] which help the model keep track of timing. Our main contributions in this paper are thus the following:

- We show that target factors can be adapted to predict durations alongside phoneme sequences to jointly optimize translation quality and speech overlap for automatic dubbing.
- We design auxiliary counters that further improve the speech overlap by providing extra information to the model to keep track of timing information.
- We evaluate our models and show that our approach improves upon previous work which instead proposed a model generating interleaved sequences of phonemes and corresponding durations.
- We release our implementation[2] and scripts[3] sufficient for replication, to enable future research in this area.

2. Related Work

Standard automatic dubbing methods [1] usually follow the pipeline where the machine translated transcript is segmented into phrases and pauses via prosodic alignment [3, 4, 5, 6], and the final output is synthesized into speech via TTS. Since this pipeline can result in output speech needing to be stretched unnaturally in order to satisfy timing constraints, some prior works have tried to avoid the separate prosodic alignment step through training models to predict pauses within translations [7], integrating isochrony constraints in MT decoding [8] or by optimizing prosody jointly with the TTS [9].

As a proxy for isochrony, some prior works have proposed optimizing isometry, i.e. generating translations which match the number of characters in the source text [10, 11], but this has been shown to be weakly correlated to isochrony [12].

Concurrent work [13] predicts word durations along with words and presents a novel loss function for decoding. They do not elaborate on how word durations are used to generate speech, and we were unable to compare results due to both their translation and TTS implementations being publicly unavailable. Other recent work [14] has presented a simple sequence-to-sequence approach to generate interleaved sequences of phonemes and corresponding duration. We follow the data/model setup and use their approach as our baseline.

Target factors have been used in statistical MT to explicitly model morphology [15]. They were adopted in neural machine translation to simultaneously translate lemmas with their corresponding parts of speech [2], and have also been shown to be effective to predict case markers [16], subword separators [17], capitalization, or gender information [18] decoupled from output tokens. In the area of isochronous MT, they have been used to predict pause markers as an alternative to generating an explicit token [7].

*This work was done during an internship at Amazon.

[1]The counters are modified target factors providing additional information to the decoder but whose outputs we do not use (§ 3).

[2]*https://github.com/awslabs/sockeye/pull/1082*

[3]*https://github.com/amazon-science/iwslt-autodub-task*

10.21437/Interspeech.2023-1063

Table 1: *An example target sequence along with its target factors. The corresponding word sequence is shown in the first row but not used in the actual model. The factors are time-shifted internally to condition factor outputs on the main output, which is not shown here. NULL is a padding token used to align the tokens correctly after the internal shift, so that the model sees* f_0^{total}, f_0^{pause}, *and* f_0^{segment} *before generating the first phoneme and duration.*

Target text		don't					you			know			[pause]	it		
f^{main}	NULL	D	OW1	N	T	⟨eow⟩	Y	UW1	⟨eow⟩	N	OW1	⟨eow⟩	[pause]	IH0	T	⟨eow⟩
f^{dur}	NULL	2	5	6	8	0	3	7	0	5	41	0	0	5	7	0
f^{total}	89	87	82	76	68	68	65	58	58	53	12	12	12	7	0	0
f^{pause}	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
f^{segment}	77	75	70	64	56	56	53	46	46	41	0	0	12	7	0	0

3. Method

We propose predicting phoneme durations as target factors [2], instead of interleaving phonemes and phoneme durations [14]. Target factors are additional output layers to produce multiple outputs at each decoder step. There are separate embedding layers for each target factor, and all the factor embeddings are concatenated to the main target embedding and provided as input to the decoder. To condition the factor outputs upon the main output, factors are shifted such that the factors corresponding to output token y_t are generated at step $t+1$. We use the Sockeye[4] target factor implementation. In contrast to the interleaved baseline [14], target factors allow us to model the phonemes and durations separately while still ensuring that they are conditioned on each other. It also significantly decreases the sequence length and eliminates the possibility of producing invalid output (e.g. two durations in a row).

In addition to the main output f^{main} and corresponding durations being generated as a target factor (f^{dur}), we propose additional input embeddings in the decoder to help the model keep track of the isochrony constraints, which we denote auxiliary counters. The counters are implemented identically to target factors (i.e. each counter has an embedding layer whose embeddings are concatenated to the target embedding[5]), except that the counters are not predicted at inference. Instead, the values of the counters are calculated at each time step based on the prior durations predicted by the model and used as input in the next step. These counters are:

- **Total frames remaining** (f_t^{total}): Keeps track of the total number of frames remaining in the sentence. This is initialized by the total desired duration of the sentence and is decremented by the phoneme duration at each output step.

$$f_t^{\text{total}} = f_{t-1}^{\text{total}} - f_t^{\text{dur}} \tag{1}$$

- **Pauses remaining** (f_t^{pause}): Keeps track of the number of pauses remaining in the sentence.

$$f_t^{\text{pause}} = \begin{cases} f_{t-1}^{\text{pause}} - 1, & \text{if } f_t^{\text{main}} = [\text{pause}] \\ f_{t-1}^{\text{pause}}, & \text{otherwise} \end{cases} \tag{2}$$

- **Segment frames remaining** (f_t^{segment}): Keeps track of the number of frames remaining in a segment, i.e. until a pause is generated, or the sentence ends. This is initialized by the segment durations from the source sentence, and is decremented by the phoneme duration at each step until a `[pause]` is generated.

$$f_t^{\text{segment}} = \begin{cases} f_{t-1}^{\text{segment}} - f_t^{\text{dur}}, & \text{if } f_t^{\text{main}} \neq [\text{pause}] \\ \text{next segment duration}, & \text{otherwise} \end{cases} \tag{3}$$

All of these auxiliary counters are calculated from the phonemes and durations in pre-processing for training, and calculated at each time step at inference time. While the model can generate predictions for counters as target factors, we only use the counters to help the model keep track of its state and discard their outputs.[6] An example of a target sequence along with its target factor and counters is shown in Table 1.

The implemented behavior of target factored models in Sockeye at inference time is to predict target factors and then feed those predictions back into the model at the next inference step. For counters (where we are trying to help the model keep track of timing), we found it critical to correctly calculate counter values according to the equations in § 3 before feeding them back to the decoder at the next time step. Compared to the default Sockeye behavior for target factors, this improved speech overlap significantly (from 0.9181 to 0.9972) without affecting translation quality.

We show in future sections that our method is able to satisfy the duration constraints almost perfectly while maintaining reasonable translation accuracy. However, in practice we do not want to achieve perfect speech overlap because it can result in poor translations or speech that is shortened/lengthened to the point where it sounds *unnatural*. In fact, analysis of human dubbing [12] has shown that human dubbers prioritize naturalness and translation quality over speech overlap.[7] For this reason, following prior work in isometric MT [20] and automatic dubbing [14], we add gaussian noise to the segment durations in our training data. This creates training examples where part or all of the translation ends slightly before or after the counters reach zero, and the model learns to be flexible with the timing information.

4. Experimental Setup

We use the English-German subset of CoVoST-2[8] as our dataset, consisting of English audio clips and transcripts along with German text translations. Each clip roughly corresponds to a sen-

[4] `https://github.com/awslabs/sockeye`

[5] As an alternative to trained embeddings for each numeric counter value, we also tried fixed sinusoidal embeddings, inspired by the positional embeddings used in transformers [19]. We found that models with sinusoidal embeddings converged faster but achieve lower translation quality. It is not clear why sinusoidal embedding would lower translation quality, and we hope to better understand (and perhaps improve on) this in future work.

[6] Additionally, our best models are trained without any gradient coming from the auxiliary counter predictions, effectively removing the part of the network predicting auxiliary counter outputs.

[7] Median overlap is just 0.731 in a large corpus of human dubs.

[8] `https://github.com/facebookresearch/covost`

tence. We run the Montreal Forced Aligner (MFA) [21] on the English audio and transcripts to get sequences of phonemes with corresponding durations. This sequence becomes our target and the German transcripts are used as the source. We mark silence of more than 0.3 seconds with `[pause]` tokens in the target phoneme sequence in order to be able to reinsert these periods of silence in final dubs. We also mark the end of words with `<eow>` tags. We calculate the duration of each segment (speech without pauses) by adding the phoneme durations between pauses, bin them into 100 bins of approximately equal frequency to avoid sparsity, and add these as tags to the source texts. We apply BPE [22] on the German text with 10k merges. Our final dataset consists of 289,074 training examples, with 15,499 examples in the validation set and 15,413 in the test set.

For all models, we use a standard transformer-base architecture, augmented with target factors and counters where applicable, trained with a maximum batch size of 32768 tokens for 600 epochs, with a dropout probability of 0.3 and label smoothing 0.1. We save checkpoints every 2000 updates and pick the best checkpoint according to COMET on the validation set.

Our baseline follows the approach described by [14], which is a simple Transformer sequence-to-sequence model. The input is the subword-level source text, with binned segment durations appended as tags to the end of the sequence and the output sequence is an interleaved sequence of phonemes and corresponding durations, with `<eow>` tags to mark the end of each word and `[pause]` tokens to mark the end of a segment. As an example, a source sentence is formatted as `Das weißt du nich@@ t@@ ? <|> <bin4> <bin1>` with the corresponding target sequence `D 2 OW1 5 N 6 T 8 <eow> Y 3 UW1 7 <eow> N 5 OW1 41 <eow> [pause] IH0 5 T 7 <eow>`.

Additionally, we train a German→English machine translation model using the same datasets at the subword level (instead of phoneme outputs), and a model to translate German text to English phoneme sequences without durations. These two models act as baselines to measure how much the translation quality deteriorates for models with duration constraints.

Since our models output sequences of phonemes, we train a Transformer seq2seq model on the same dataset to transform English phoneme sequences into sequences of English words. Translation quality is then evaluated using BLEU[9] [23, 24], Prism [25, 26], and COMET[10] [27]. We find the metrics to be highly correlated in our results, and thus report only BLEU scores.

To quantify speech overlap between the reference (ref.) and the hypothesis (hyp.), we use the relative difference of duration between reference segments and predicted translated segments, averaged over all segments in the dataset:

$$\text{Speech Overlap} = 1 - \frac{|\text{ref. duration} - \text{hyp. duration}|}{\text{ref. duration}} \tag{4}$$

As an additional automatic metric, we also count the number of sentences in the validation and test sets where the wrong number of pauses is generated.

5. Analysis

We train models to predict phoneme durations as a target factor and use all the auxiliary counters described in § 3.

[9] SacreBLEU: BLEU|#1|c:lc|e:no|tok:none|s:exp|v:2.3.1
[10] wmt20-comet-da

Table 2: *Summary of key results for some representative models on the test set. The models with all the target counters use the optimal configuration from § 5.*

Model Configuration	BLEU ↑	Speech Overlap ↑
Text to text (MT)	38.0	–
Text to phonemes	35.8	–
Interleaved, no noise	32.0	0.8702
Interleaved, noised 0.2	35.4	0.7105
Single target factor, no noise	33.8	0.8931
+ all counters, no noise	34.0	0.9887
+ all counters, noised 0.1	35.6	0.8649

Embedding Size: Since the factored architecture adds a large number of parameters in the form of embedding matrices and output layers to the model, we want to optimize the factor embedding size so that it is large enough to adequately represent all the possible factor/counter values while not being too large to train in our limited data scenario. We sweep through a range of embedding sizes (Figure 1) and find that 64 dimensions is an optimal size. We set the embedding size for the f^{pause} counter to half of the other counter embeddings since it has much fewer possible values than the other counters.

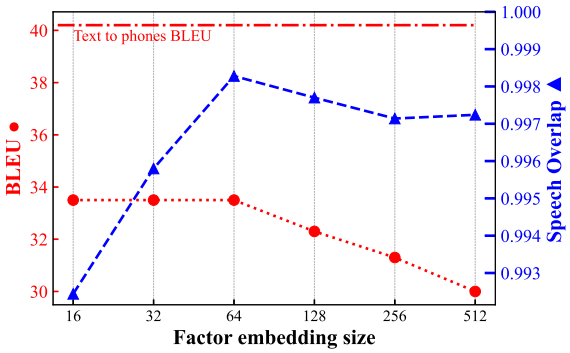

Figure 1: *Results on the validation set varying the factor and counter embedding sizes. f^{pause} embedding size is always half of the other counters. Models trained with equal loss weights on factors and counters, on data with clean segment durations.*

Counter Loss Weights: At training time, counters are predicted at each step just like target factors, and all the factors/counters are assigned an equal weight for loss computation by default, i.e. the cross-entropy losses for the output and all target factors are simply summed. However, it is possible to generalize the loss by assigning different weights to the outputs. Since we do not use the outputs for the counters, we can set the weights of the counters to 0, thus letting the model focus on the phoneme and duration outputs that we actually need. We find that zeroing the loss weights of the counters helps improve translation quality by 4.2 BLEU at the cost of a very small drop (0.003) of speech overlap.

Adding Noise: We find that as we add more noise, for both the interleaved as well as factored models, the translation quality increases at the cost of speech overlap, ultimately matching the text-to-phonemes baseline. (Figure 2). The amount of noise allows us to control the trade-off between translation quality and speech overlap.

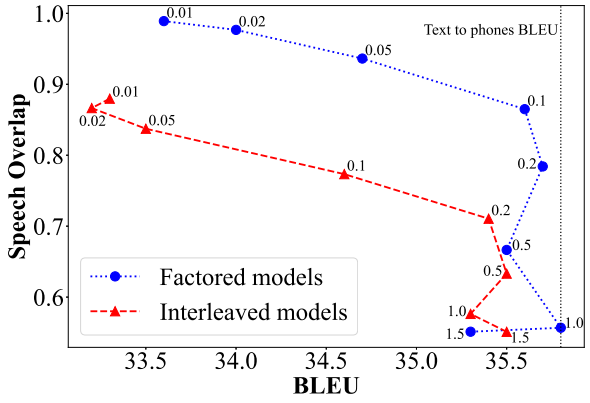

Figure 2: *Variation of translation quality (BLEU) and speech overlap with different amounts of noise added to the segment durations. Results shown on the test set. Each point annotation indicates the standard deviation of the added noise.*

Counter Ablations: To evaluate the effectiveness of the counters and source duration tags, we start with the highest-quality factored model – embedding sizes 64, 64, 64, 32 with zeroed counter loss weights – and measure the change in translation accuracy and speech overlap for models with one or more of the counters removed. From Table 3, we can see that removing either the source segment tags or f^{total} has very little impact on the speech overlap, since the model is able to track the timing information from f^{segment}. Removing both the source segment tags and f^{segment} causes a large drop in speech overlap since the model has no information about segment durations. We also see that removing f^{segment} and f^{pause} causes a large number of outputs to have the wrong number of pauses.[11] These results are consistent with our intuition about the purpose of each of these counters.

Table 3: *Counter ablation results on the test set. We remove the auxiliary counters and/or source segment durations and measure the effect. W.P. represents the number of sentences in the test set for which the wrong number of pauses is generated.*

Model configuration	BLEU ↑	Overlap ↑	W.P. ↓
All counters + source durations	34.0	0.9887	29
Without:			
Source durations	33.4	0.9900	25
f^{total}	34.0	0.9914	8
f^{segment}	34.0	0.9258	109
f^{segment} + f^{pause}	33.9	0.9294	176
Source durations + f^{segment}	34.1	0.6214	103
f^{total} + f^{segment}	33.9	0.9191	20

6. Results

The translation quality of the text-to-phones baseline is 2.2 BLEU lower than the text-to-text (i.e. standard MT) model (see Table 2). This is likely due to: (1) To evaluate the text-to-phoneme model, we are mapping phonemes to words using a seq2seq model and then scoring with word-level metrics. The seq2seq model is not perfect and is likely introducing some errors, making the text-to-phoneme model appear to be worse than it actually is, and (2) We used the same parameters for the phoneme model as the text model. We did not attempt to optimize the transformer parameters for phonemes, but plan to do so in future work.

Modeling phoneme durations using target factors improves both translation quality (+1.8 BLEU) and speech overlap (+0.023) relative to the interleaved baseline (see Table 2, no noise settings).

Adding auxiliary counters provides nearly perfect speech overlap (0.9887, perfect score is 1.0) in the no noise setting. It provides substantial improvement in speech overlap (+0.0956) compared to the target factor model without auxiliary counters, while marginally improving translation quality (+0.2 BLEU) (see Table 2, no noise settings).

By adding noise to the speech segment durations, we are able to obtain nearly the same translation quality as the text-to-phoneme model (35.6 vs 35.8) while still achieving very high speech overlap (0.8649, higher than observed in human dubs).

7. Qualitative Perception Results

We intended to perform human evaluation of dubbed videos using crowd source workers but a pilot showed very noisy results, with annotators often appearing to ignore annotations guidelines. We believe this is due at least in part to the large number of factors that affect perception of a dubbed video, including (but not limited to) translation quality, speech quality / naturalness, isochrony, and lip sync.

We present instead some qualitative conclusions drawn by the authors after watching/listening to many samples. The baseline tends to be the most natural sounding, but the lack of isochrony is disconcerting.[12] The proposed models with little or no noise added have much better isochrony, as expected, but often sound a little more robotic than the baseline, and it is not unusual to have a word at the end of a segment repeated (presumably this happens when the translation model finishes a translation but the counters tell the model it must keep producing output). The proposed models with large amounts of noise also sounded a bit unnatural, but for a very different reason. The speech in the test set appears to be fairly slow compared to the training data, while the model produces speech with speaking rates similar to the training data, resulting in speech segments which are often short, resulting in long, often unnatural pauses between speech segments. The proposed models with with noise of around 0.1 seem to be the best compromise between isochrony and naturalness/translation quality, consistent with the automatic evaluation (see Figure 2).

8. Conclusions

In this paper, we have shown that target factors can be used to predict phoneme durations alongside translated phoneme sequences to jointly optimize translation and timing for automatic dubbing. We train models with target factors for duration prediction as well as other auxiliary counters to further guide the model. Automatic evaluation show that our models out-perform a baseline of training a model to generate interleaved phoneme and duration sequences.

[11] We cannot remove only f^{pause} since f^{segment} uses f^{pause} to fetch the correct segment durations.

[12] We believe the lack of isochrony would be even more jarring when viewing dubbed content with multiple speakers.

9. References

[1] M. Federico, R. Enyedi, R. Barra-Chicote, R. Giri, U. Isik, A. Krishnaswamy, and H. Sawaf, "From speech-to-speech translation to automatic dubbing," in *Proceedings of the 17th International Conference on Spoken Language Translation*. Online: Association for Computational Linguistics, Jul. 2020, pp. 257–264. [Online]. Available: https://aclanthology.org/2020.iwslt-1.31

[2] M. García-Martínez, L. Barrault, and F. Bougares, "Factored neural machine translation architectures," in *Proceedings of the 13th International Conference on Spoken Language Translation*. Seattle, Washington D.C: International Workshop on Spoken Language Translation, Dec. 8-9 2016. [Online]. Available: https://aclanthology.org/2016.iwslt-1.3

[3] A. Öktem, M. Farrús, and A. Bonafonte, "Prosodic Phrase Alignment for Machine Dubbing," in *Proc. Interspeech 2019*, 2019, pp. 4215–4219.

[4] M. Federico, Y. Virkar, R. Enyedi, and R. Barra-Chicote, "Evaluating and Optimizing Prosodic Alignment for Automatic Dubbing," in *Proc. Interspeech 2020*, 2020, pp. 1481–1485. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2983

[5] Y. Virkar, M. Federico, R. Enyedi, and R. Barra-Chicote, "Improvements to prosodic alignment for automatic dubbing," in *ICASSP 2021*, 2021. [Online]. Available: https://www.amazon.science/publications/improvements-to-prosodic-alignment-for-automatic-dubbing

[6] ——, "Prosodic alignment for off-screen automatic dubbing," in *Interspeech 2022*, 2022. [Online]. Available: https://www.amazon.science/publications/prosodic-alignment-for-off-screen-automatic-dubbing

[7] D. Tam, S. M. Lakew, Y. Virkar, P. Mathur, and M. Federico, "Isochrony-aware neural machine translation for automatic dubbing," in *Interspeech 2022*, 2022. [Online]. Available: https://www.amazon.science/publications/isochrony-aware-neural-machine-translation-for-automatic-dubbing

[8] A. Saboo and T. Baumann, "Integration of dubbing constraints into machine translation," in *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 94–101. [Online]. Available: https://aclanthology.org/W19-5210

[9] C. Hu, Q. Tian, T. Li, W. Yuping, Y. Wang, and H. Zhao, "Neural dubber: Dubbing for videos according to scripts," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[10] S. M. Lakew, Y. Virkar, P. Mathur, and M. Federico, "Isometric mt: Neural machine translation for automatic dubbing," in *ICASSP 2022*, 2022. [Online]. Available: https://www.amazon.science/publications/isometric-mt-neural-machine-translation-for-automatic-dubbing

[11] S. M. Lakew, M. Federico, Y. Wang, C. Hoang, Y. Virkar, R. Barra-Chicote, and R. Enyedi, "Machine translation verbosity control for automatic dubbing," in *ICASSP 2021*, 2021. [Online]. Available: https://www.amazon.science/publications/machine-translation-verbosity-control-for-automatic-dubbing

[12] W. Brannon, Y. Virkar, and B. Thompson, "Dubbing in Practice: A Large Scale Study of Human Localization With Insights for Automatic Dubbing," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 419–435, May 2023. [Online]. Available: https://doi.org/10.1162/tacl_a_00551

[13] Y. Wu, J. Guo, X. Tan, C. Zhang, B. Li, R. Song, L. He, S. Zhao, A. Menezes, and J. Bian, "Videodubber: Machine translation with speech-aware length control for video dubbing," 2022. [Online]. Available: https://arxiv.org/abs/2211.16934

[14] A. Chronopoulou, B. Thompson, P. Mathur, Y. Virkar, S. M. Lakew, and M. Federico, "Jointly Optimizing Translations and Speech Timing to Improve Isochrony in Automatic Dubbing," Feb. 2023, arXiv:2302.12979.

[15] O. Bojar, "English-to-czech factored machine translation," in *Proceedings of the second workshop on statistical machine translation*, 2007, pp. 232–239.

[16] M. Nădejde, S. Reddy, R. Sennrich, T. Dwojak, M. Junczys-Dowmunt, P. Koehn, and A. Birch, "Predicting target language CCG supertags improves neural machine translation," in *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 68–79. [Online]. Available: https://aclanthology.org/W17-4707

[17] P. Wilken and E. Matusov, "Novel applications of factored neural machine translation," *arXiv preprint arXiv:1910.03912*, 2019.

[18] X. Niu, G. Dinu, P. Mathur, and A. Currey, "Faithful target attribute prediction in neural machine translation," *arXiv preprint arXiv:2109.12105*, 2021.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[20] P. Wilken and E. Matusov, "AppTek's submission to the IWSLT 2022 isometric spoken language translation task," in *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Dublin, Ireland (in-person and online): Association for Computational Linguistics, May 2022, pp. 369–378. [Online]. Available: https://aclanthology.org/2022.iwslt-1.34

[21] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Proc. Interspeech 2017*, 2017, pp. 498–502.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162

[23] 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 of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[24] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 186–191. [Online]. Available: https://aclanthology.org/W18-6319

[25] B. Thompson and M. Post, "Automatic machine translation evaluation in many languages via zero-shot paraphrasing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 90–121. [Online]. Available: https://aclanthology.org/2020.emnlp-main.8

[26] ——, "Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity," in *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 561–570. [Online]. Available: https://aclanthology.org/2020.wmt-1.67

[27] R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, "COMET: A neural framework for MT evaluation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2685–2702. [Online]. Available: https://aclanthology.org/2020.emnlp-main.213