Morphologically Motivated Input Variations and Data Augmentation in Turkish-English Neural Machine Translation

ZEYNEP YİRMİBEŞOĞLU and TUNGA GÜNGÖR, Boğaziçi University

Success of neural networks in natural language processing has paved the way for neural machine translation (NMT), which rapidly became the mainstream approach in machine translation. Significant improvement in translation performance has been achieved with breakthroughs such as encoder-decoder networks, attention mechanism, and Transformer architecture. However, the necessity of large amounts of parallel data for training an NMT system and rare words in translation corpora are issues yet to be overcome. In this article, we approach NMT of the low-resource Turkish-English language pair. We employ state-of-the-art NMT architectures and data augmentation methods that exploit monolingual corpora. We point out the importance of input representation for the morphologically rich Turkish language and make a comprehensive analysis of linguistically and non-linguistically motivated input segmentation approaches. We prove the effectiveness of morphologically motivated input segmentation for the Turkish language. Moreover, we show the superiority of the Transformer architecture over attentional encoder-decoder models for the Turkish-English language pair. Among the employed data augmentation approaches, we observe back-translation to be the most effective and confirm the benefit of increasing the amount of parallel data on translation quality. This research demonstrates a comprehensive analysis on NMT architectures with different hyperparameters, data augmentation methods, and input representation techniques, and proposes ways of tackling the low-resource setting of Turkish-English NMT.

CCS Concepts: • **Computing methodologies** → **Machine translation**; **Neural networks;**

Additional Key Words and Phrases: Neural machine translation, morphology, low-resource, Transformer, encoder-decoder, attention, data augmentation, word segmentation

ACM Reference format:
Zeynep Yirmibeşoğlu and Tunga Güngör. 2023. Morphologically Motivated Input Variations and Data Augmentation in Turkish-English Neural Machine Translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 22, 3, Article 92 (March 2023), 31 pages.
https://doi.org/10.1145/3571073

1 INTRODUCTION

Overcoming language barriers between people has been a concern of humankind for ages. Communication between people that speak different languages and availability of literary or professional text are achieved through human translation. However, having access to high-quality human translation and widespread use of it is, to this day, a costly issue. Unavailability and expensiveness of

Authors' address: Z. Yirmibeşoğlu and T. Güngör, Boğaziçi University, Computer Engineering, Sarıyer, İstanbul, Turkey, 34342; emails: {zeynep.yirmibesoglu, gungort}@boun.edu.tr.

2375-4699/2023/03-ART92
https://doi.org/10.1145/3571073

human translation and advances in computer science and natural language processing have led to the idea of automatic translation of languages: machine translation.

Adoption and success of deep learning and neural networks in natural language processing has been a tremendous step in machine translation history, initiating the work on **neural machine translation (NMT)**. By modeling the entire machine translation system as an end-to-end neural network and eliminating excessive feature engineering, NMT gradually replaced **statistical machine translation (SMT)**, becoming the new state of the art. Significant breakthroughs have been achieved in NMT with the introduction of the encoder-decoder network, attention mechanism, and Transformer architecture. Even though the encoder-decoder and Transformer architectures effectively extract the syntactic and semantic information from a bitext, the lack of large amounts of parallel data for training an NMT system has become one of the most investigated issues. Data augmentation methods for low-resource scenarios and powerful input representation approaches for the open-vocabulary problem have been proposed, taking NMT one step further.

In this study, Turkish-English NMT is investigated. This is an especially challenging task due to the notable dissimilarity and the low-resource setting of the Turkish-English language pair. Turkish is a language with complex morphology, which causes the extraction of information from unsegmented words to be rather troublesome.

To tackle this difficult task, we provide a comprehensive analysis on state-of-the-art NMT model architectures, data augmentation techniques, and input segmentation methods for Turkish-English machine translation. We make use of the attentional encoder-decoder model with deep transition and BiDeep architectures, and the Transformer architecture, pressing the importance of model and hyperparameter selection in Turkish-English NMT. We conduct an exhaustive survey on the data sparsity issue in NMT, resulting in the selection of three data augmentation approaches for this task: self-training, back-translation, and copied data. These approaches are exploited to expand the training corpus size from 207K sentences up to 6.9M sentences, observing the benefits of each approach separately and together.

Contributions of this study can be listed as follows:

- We introduce nine morphologically motivated input segmentation methods for the Turkish language, in comparison to two of the most widely used non-morphologically motivated input representation approaches. After extensive experimentation, we show the advantages of employing linguistically motivated input representations in Turkish-English NMT, in addition to an analysis of the strengths and weaknesses of each input variation.
- We point out the dominance of the Transformer architecture over the attentional encoder-decoder architecture with respect to translation quality in the Turkish-English NMT task.
- We make use of and compare three data augmentation techniques (self-training, back-translation, and copying monolingual data), revealing back-translation to be the most effective one in this low-resource setting.

Our code has been released online.[1]

This article is organized as follows. Section 2 gives a comprehensive literature review on NMT. The datasets and statistics about the datasets used in this work are given in Section 3. Section 4 describes the model architectures and the data augmentation and input segmentation methods. Section 5 explains the ensembles of the models used. Experimental results and their analyses are provided in Section 6. Finally, the work is summarized and future work is suggested in Section 7.

[1] https://github.com/zeynepyirmibes/Morphologically-Motivated-TR-EN-NMT.

2 RELATED WORK

Among several machine translation approaches, including rule-based, statistical, example-based, and neural machine translation, this research focuses on the most recent methods that lie around the NMT approach. In this section, we review the works on NMT with an emphasis on data augmentation approaches to tackle the low-resource scenario in NMT. The Turkish-English news translation tasks in EMNLP's 2017 and 2018 Conferences on Machine Translation (WMT17, WMT18) in particular pose a rich variety of models for Turkish-English NMT, pressing the importance of data augmentation in this low-resource setting.

The input of an NMT system can make all the difference. The morphologically rich characteristic of the Turkish language has urged researchers to focus on more morphologically motivated inputs. Therefore, we also investigate the most frequently used input representations and linguistically inspired input variations.

2.1 Neural Machine Translation

The introduction of neural networks into the realm of machine translation can be traced back to late 1990s with the works of Forcada and Ñeco [20] and Castaño et al. [10], which could not be further investigated due to inadequate computational resources at that time. In 2013, Kalchbrenner and Blunsom [34] introduced the **recurrent neural networks (RNN)** for translation modeling, laying the foundation of NMT. After this breakthrough, sequence-to-sequence NMT models started to emerge mostly in the form of an encoder-decoder architecture, where the source sentence is encoded into a fixed-length vector from which the decoder generates the target sentence [14, 61]. The introduction of the encoder-decoder model is an important milestone in NMT. Addressing the issue of translating long sentences, the encoder-decoder model was further enhanced with the addition of Bahdanau attention [4] and global/local attention mechanisms [40].

An issue put forward by Liu et al. [37] is unbalanced outputs in RNN-based NMT (RNMT), arising from large vocabularies, frequent reordering between input and output sentences, and long sentences. A solid example of this phenomenon is shown in their analysis on Japanese-English translation hypotheses, where the translation quality of the prefixes of the source sentence (i.e., initial words of the sentence) is much higher than that of the suffixes (i.e., last words of the sentence) for **left-to-right (L2R)** decoding. As a solution, they generate hypotheses from **right-to-left (R2L)** in addition to L2R and enforce target agreement of these separate models via joint search, producing more balanced outputs and conserving good translation quality for both prefixes and suffixes. Bidirectional decoding has been further employed through rescoring n-best translation hypotheses [55], inference with linear relaxation [26], neural forward (for L2R models) and backward (for R2L models) decoders for asynchronous bidirectional inference [72], and a single bidirectional decoder for synchronous bidirectional inference [75]. Latest models prefer beam or greedy search for translation [72, 75].

The success of the attention mechanism brought with it the idea of self-attention, where attention is used not only between the encoder and the decoders but within them. Vaswani et al. [65] introduced two new self-attention mechanisms (scaled dot-product and multi-head attention) and a new architecture called *Transformer* relying completely on self-attention to deduct the global relationships between input and output. With this new architecture and the semisupervised method of back-translation as a way of incorporating monolingual data, state-of-the-art results have been reached for the WMT14 English-German test set [19].

Developing deep NMT models with better performance has received tremendous attention from researchers, resulting in advanced NMT models that are variants of the vanilla Transformer and the attentional encoder-decoder. RNMT+ [12], an enhancement over Google's RNN-based GNMT (Google's NMT) model [70], consisted of six bidirectional LSTMs in the decoder and took advantage of the Transformer model's multi-head additive attention. Deep Transformer models, such as a 16-layer model with transparent attention [5] and a Transformer model with a 30-layer encoder with layer normalization [66], have shown great promise, whereas a simplified architecture with comparable performance can also be achieved by applying neural architecture search [60].

2.2 Data Augmentation

Sparsity of sentence-aligned parallel corpora significantly degrades the performance of NMT systems for low-resource language pairs. To tackle this issue, ways of extracting and exploiting the linguistic knowledge within monolingual corpora, which are much more accessible, have been investigated by researchers. One of the first works that incorporated monolingual data into their NMT system is Gülçehre et al. [76], where they integrated RNN language models trained on monolingual target-side data. Another method presented itself in WMT15 through rescoring the n-best hypotheses of the NMT model with n-gram LMs [31].

Sennrich et al. [56] introduced two strategies to leverage monolingual data: empty (dummy) source sentences and synthetic source sentences. The former requires parallel examples with an empty source-side, implying the context vector to be uninformative, enforcing the network to learn solely from previous target words. The latter is the novel *back-translation* approach, which is the automatic translation of monolingual target data into synthetic source data. In this case, the target-side is authentic monolingual text, and only the source-side is synthetic. After obtaining dummy or back-translated source data, NMT networks are trained with a mixture of parallel data and these pseudo parallel data.

The back-translation approach has been further investigated, revealing an improvement of translation performance with larger amounts of back-translated data, until the point where the imbalance is too much in favor of the synthetic data [47]. Iterative back-translation [27] and using back-translation in hierarchical transfer learning [38] have improved generalization with respect to baseline back-translation methods. Other enhancements over the original back-translation method include sampling multiple source sentences based on word distribution of output words [28] or sampling a single source sentence in addition to adding noise to beam search outputs [19], showing improvement in translation accuracy. Caswell et al. [11] revealed the role of noise in back-translation, which turned out to be helping the model distinguish between original and synthetic data. Another way of distinguishing between authentic and synthetic data to improve back-translation is through uncertainty-based confidence measures [67].

Improvement in translation quality that comes with data augmentation through original target-side monolingual data has given birth to another strategy: copied monolingual data [16]. This technique involves copying the target-side monolingual data to the source-side, creating a bitext with each source sentence identical to the target sentence. Afterward, the copied data is mixed with the original parallel corpus to form the final training set.

Another strategy for data augmentation is self-training, where source-side monolingual data is translated to the target-side and then used as additional parallel data [71]. He et al. [25] more recently revisited self-training with injected noise, observing once again its smoothing effect. The work of Jiao et al. [32] asserts that self-training significantly improves the translation quality of uncertain sentences, especially for low-frequency words.

Works that incorporate both source-side and target-side monolingual corpora have also shown great promise. Among these works are adopting a strategy to leverage both sides [69], an autoencoder that reconstructs the observed monolingual corpora [13], reinforcement learning with source and target-side LMs [24], iterative back-translation [27], and a mirror-generative NMT that can learn from the monolingual corpora by jointly training source-target and target-source NMT models and two language models [73].

Table 1. Turkish-English News Translation Results on the WMT17 Test Set

System	Model	Input	Monolingual Data	BLEU
LIUM [21]	Attentional encoder-decoder	BPE	150K back-translated	17.91
AFRL-MITLL [22]	Attentional encoder-decoder	BPE	14M back-translated	18.05
UEDIN (2017) [53]	Stacked Attentional encoder-decoder	BPE	400K back-translated + 400K copied	20.1
UEDIN (2018) [23]	Transformer	BPE	2.5M back-translated + 1M copied	26.6

A comprehensive analysis on the effects of hyperparameters on the low-resource setting has shown that reducing the **byte pair encoding (BPE)** vocabulary size, using word dropout, and tuning the hyperparameters are extremely important performance boosters [58]. In that work, the domination of NMT over phrase-based SMT in the low-resource setting for far less parallel training data has also been confirmed.

2.3 WMT17 and WMT18 Tasks

The WMT17 News Translation Task is a shared task that entails the Chinese-English, Czech-English, Finnish-English, German-English, Latvian-English, Russian-English, and Turkish-English language pairs. A total of 103 submissions from 31 institutions were made [8]. Seven systems (four SMT and three NMT) have been submitted for the Turkish-English direction. In this study, the three Turkish-English NMT systems in WMT17 and their performances are taken into account (Table 1). All reported BLEU scores are of official submissions in WMT17, except for UEDIN's improved result in 2018 for the WMT17 test set.

Due to the low-resource characteristic of the Turkish-English language pair (approximately 220K parallel sentences in the SETimes corpus) and the need for exploiting largely available monolingual data, all NMT systems with submissions in Turkish-English have used back-translation, approaching this technique in different ways.

The LIUM system in WMT17 used a bidirectional **Gated Recurrent Unit (GRU)** encoder with layer normalization and a conditional GRU (cGRU) decoder with attention, employing tied embeddings (for feedback and output embeddings) [21]. The back-translated data amount was kept at around 150K sentences to abide with the original-to-synthetic ratio. They also experimented with different amounts of back-translated data in the English-Turkish direction, observing that the original-to-synthetic ratio can be disregarded and the increase in the back-translated data amount is significantly beneficial. A 4.6 BLEU score improvement was achieved with 1M sentences as opposed to 150K sentences where the original-to-synthetic ratio was preserved.

The AFRL-MITLL system in WMT17 employed an iterative approach for back-translation [22]. The initial model was a Turkish-English SMT model trained with Moses [36]. Afterward, an English-Turkish Marian system [33] was trained on the parallel data and the back-translated data from the Moses model, which was used to translate the English monolingual data (around 9 million sentences). Finally, two L2R Marian models and one R2L Nematus model [54] were trained on the parallel data and the back-translated data from the previous Marian model. The final translation was an ensemble of two L2R Marian models, rescored by the R2L Nematus model.

Table 2. Turkish-English News Translation Results (Official) on the WMT18 Test Set

System	Model	Input	Monolingual Data	BLEU
NICT [42]	Transformer	BPE	1.6M back-translated	26.9
UEDIN [23]	Transformer	BPE	2.5M back-translated + 1M copied	26.9

The University of Edinburgh (UEDIN) system in WMT17 [53] used a stacked attentional encoder-decoder architecture proposed by Zhou et al. [74], where the LSTM layers are stacked and residual connections are used between stack layers. They leveraged back-translation and a copied monolingual corpus. The final training corpus consisted of parallel, copied, and back-translated data with a 1:2:2 ratio. An ensemble of four L2R Nematus models was used for obtaining the 50 best translation hypotheses, which were in turn rescored by an ensemble of four Nematus R2L models. The ensemble model received a cased BLEU score of 20.1, the highest among the submitted systems.

The WMT18 News Translation Task entails the Chinese-English, Czech-English, Estonian-English, Finnish-English, German-English, Kazakh-English, Russian-English, and Turkish-English language pairs, receiving 103 submissions from 32 institutions [9]. The results of the official submissions of the two systems for the Turkish-English direction are given in Table 2.

The NICT system in WMT18 [42] incrementally trained their Marian Transformer models, increasing the amount of their back-translated data at each iteration, finally reaching 1.6M back-translated sentences. They combined their phrase-based SMT system with the NMT system by generating 100-best translation hypotheses and rescored them using a reranking framework. Their combined system received a cased BLEU score of 26.9 for the *newstest2018* test set.

The UEDIN system in WMT18 [23] employed the Transformer architecture and a deep RNN architecture, both of which were implemented using the Marian tool. The deep RNN was described as a BiDeep GRU encoder-decoder [43], which was used with multi-head and multi-hop attention. Using the deep RNN setting, a back-translation system was trained using only the 200K parallel corpus. Using this model, 800K sentences were back-translated, creating a second back-translation system with the combination of the parallel corpus and the synthetic corpus (1M sentences). Afterward, 2.5M sentences were back-translated with the second deep RNN model. For the final Marian Transformer models, one setting of the training corpus was the 2.5M synthetic sentences in addition to the parallel corpus oversampled five times (1M sentences). The second setting was the previous setting with the addition of 1M copied data, obtained the same way as in the WMT17 task. Six independently trained L2R models (checkpoints that achieve the best BLEU score during training) were used for translation and three R2L for rescoring with a beam size of 20, yielding an official 26.9 BLEU score for the *newstest2018* test set. Their best result was obtained from the same six L2R and three R2L models with a beam size of 30, receiving 28.2 BLEU (for *newstest2018*) after the shared task submission, which we report as the state-of-the-art result in Section 6.4. They also improved their state-of-the-art submission for the WMT17 shared task with the same system and beam size (30), obtaining 26.6 BLEU for the *newstest2017* test set (see Table 1).

All aforementioned models in the WMT17 and WMT18 tasks used BPE [57] as the input scheme with the *subword-nmt* tool [49].

2.4 Input Variations

Large vocabularies and **out-of-vocabulary (OOV)** words have been the focus of researchers in NMT due to the open vocabulary setting of neural translation. To cope with the increase in training complexity due to large target vocabularies, Jean et al. [30] proposed importance sampling by exploiting a small subset of the vocabulary. Other techniques include a post-processing step that looks up OOV words from a dictionary [41] and the representation of OOV words as character embeddings [39]. Addressing both the OOV and the morphologically complex word problems, Sennrich et al. [57] proposed a word segmentation scheme called *BPE*. In this scheme, words are divided into subword units from a set of frequent pairs of characters. Their method allows a fixed-size vocabulary and the ability to represent OOV or morpholocially complex words efficiently.

The morphologically rich characteristic of Turkish requires particular attention in the translation task. Being a highly agglutinative language, multiple morphemes can be concatenated posing a large variety of inflections and derivations such that a single word in Turkish may and often does correspond to multiple words in English. An example is *okulundaydı*, which can be translated as: "He/she was at his/her school." The correct segmentation of this word would be *okul (school) + u (his/her) + nda (at) + ydı (he/she was)*. Thus, the significance of input decomposition for the Turkish-English NMT task comes to surface, and we can expect better translation quality if the correct segmentation of morphemes inside a Turkish word is achieved.

Gülçehre et al. [76] employed an encoder-decoder model with Bahdanau attention, in which the input of the NMT model was prepared by tokenizing the Turkish sentences using the Zemberek tool [1], then employing morphological analysis and disambiguation using the tool of Sak et al. [50], and finally removing non-surface morphemes (part-of-speech tags, etc.). The same pre-processing approach was employed by Shen et al. [59] in their densely connected NMT system. Sennrich et al. [56] relied on the same architecture and the same pre-processing for Turkish sentences as Gülçehre et al. [76], where they also incorporate back-translation.

Bektaş et al. [7] tokenized the Turkish sentences using the Moses tokenizer [36], followed by the morphological analyzer of Oflazer [44] and the morphological disambiguator of Sak et al. [50] to produce the Turkish input representation for their Turkish-English SMT system. They only kept the morphological features that correspond to lexical morphemes inside the word (dative, accusative, past participle, etc.) for the input segmentation of the word. Ataman et al. [2] also followed the same pre-processing approach but included the root and all suffix tags in the Turkish input representation of their NMT model.

Pan et al. [46] proposed a multi-source neural model with two encoders, namely a word-based encoder for source word features and a knowledge-based encoder for source morphological features. The morphological features entail the lemma, part-of-speech tag, and the morphological tag. They used BPE for segmentation, followed by the Zemberek tool [1] and the morphological disambiguator of Sak et al. [50].

3 DATASETS

Table 3 lists the parallel and monolingual corpora used in this study and the statistics about those corpora. The SETimes (Southeast European Times) corpus is a parallel corpus gathered from news articles in 10 Balkan languages, containing 45 bitexts [63, 64]. In this work, we use the Turkish-English SETimes parallel corpus that consists of 207K sentences. We tokenized and cleaned the sentences (sentences with less than 1 and more than 80 tokens) using the Moses cleaning scripts [36] before truecasing and further word segmentation. We named the pre-processed corpus *SETimes-clean* and use it for training.

For monolingual data, we utilize the WMT News Crawl 2020 dataset [6]. The dataset has been extracted from online newspapers, sentence-split, shuffled, and released for the WMT shared tasks. The Turkish monolingual corpus consists of 26,552,319 sentences, and the English corpus consists of 274,929,980 sentences. We have only used portions of the monolingual data due to high computational requirements of training NMT models and to preserve a reasonable balance between the

Table 3. Corpus Statistics

			Turkish		English	
Corpus	**Usage**	**Sentences**	**Tokens**	**Unique Tokens**	**Tokens**	**Unique Tokens**
SETimes	–	207,678	4,655,869	168,036	5,237,327	70,573
SETimes-clean	Train	207,373	4,633,304	167,519	5,210,932	70,356
newstest2016	Dev	3,000	54,420	16,441	67,468	9,700
newstest2017	Test	3,007	55,527	15,777	68,739	9,466
newstest2018	Test	3,000	57,377	17,141	70,575	10,109
WMT News Crawl (TR)	Aug.	3,502,414	58,146,344	997,387	–	–
WMT News Crawl (EN)	Aug.	3,409,247	–	–	92,807,980	591,787

number of augmented data and original parallel data. Only the used portions of the corpora are reported in Table 3, and the usage is denoted as "Aug." for data augmentation.

For all models, the WMT16 test set (*newstest2016*) is used for validation (development), and the WMT17 (*newstest2017*) and WMT18 (*newstest2018*) test sets are used for testing, taking an example from Haddow et al. [23].

4 METHODOLOGY

4.1 Encoder-Decoder Model

The encoder-decoder architecture can be considered as a dominating architecture in NMT, where RNNs are used for sequence-to-sequence prediction. The main purpose here is to extract a fixed-length vector from a variable-length input sentence and then generate a variable-length target sentence. We employ the encoder-decoder model with Bahdanau attention [4], this way aligning and translating at the same time to better translate long sentences.

In this study, the Marian toolkit [33] is used for all experiments due to its state-of-the-art results in WMT17 and WMT18 for Turkish-English and additional benefits such as high training and translation speed and multi-GPU training. Marian's attentional encoder-decoder is equivalent to that of Nematus [54], which follows the architecture proposed by Bahdanau et al. [4]. The deep encoder-decoder architectures implemented in Marian are explained in the following sections.

4.1.1 Deep Transition Architecture. The deep transition RNN employs multiple transition layers of GRU blocks, connected in such a way that the state output of one is the state input of the next one. Recurrence is implemented at the level of the whole multi-layer recurrent cell instead of individually at each GRU transition. Application of this architecture to NMT is a novel contribution of Miceli Barone et al. [43].

In this research, the Marian implementation of the deep transition architecture with an encoder recurrence depth of $L_s = 4$ and a decoder recurrence depth of $L_t = 8$ is adopted in all of the attentional encoder-decoder experiments except the final models (Sections 6.4 and 6.5). The embedding size and the hidden state size are set to 512 and 1024, respectively. Tied embeddings (weight tying of all embeddings and output layer) [48] are employed to reduce the number of parameters. To reduce training time, layer normalization [3] as an alternative to batch normalization is used. Different from batch normalization, layer normalization operates on the channel dimension instead of the batch dimension, computing the normalization statistics from the summed inputs to the neurons within a hidden layer, hindering new dependencies within the training cases. Layer normalization is applied to all recurrent and feed-forward layers with the exception of layers followed by a softmax. A dropout of 0.1 is applied along the RNN layers.

Taking an example from UEDIN's WMT18 system [23], Adam [35] is used for the optimization of the models with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The learning rate is started at 0.0003 during training. Exponential smoothing, gradient clipping, and, for regularization, label smoothing [62] (0.1) as a way of encouraging the model to be less confident are incorporated.

The models have been trained on two GPUs on the TÜBİTAK ULAKBİM's computing infrastructure TRUBA (Turkish National e-Science e-Infrastructure) with a mini-batch size fit into 9.5 GB of GPU memory. Early stopping with a patience of 5 has been selected as the stopping criterion with word-level cross-entropy used as the validation metric every 5,000 updates, up to 8 or 12 epochs. Training time differs according to the size of the training corpus and the convergence of the model. The best models according to the BLEU score for the validation set have been kept.

4.1.2 Stacked Architecture. The stacked attentional encoder-decoder architecture is not used directly in this research. However, it is explained here for the sake of the BiDeep architecture, which is a combination of deep transitions and stacking.

The stacked architecture is a GRU-based NMT model with residual connections between the stack layers. Multiple connected GRUs run for the same number of steps. At each timestep, the bottom GRU takes external inputs from the outside, whereas the higher GRUs are fed as external input the state output of the one below them. Information flow is improved with residual connections between states at different depths. The main difference from the deep transition architecture is the individual recurrence within each GRU transition block [43].

4.1.3 BiDeep Architecture. The BiDeep RNN is a novel architecture proposed by Miceli Barone et al. [43] as a mixture of deep transition and stacked architectures. Individually recurrent GRUs of the stacked encoders and decoders are replaced with multi-layer deep transition cells consisting of GRU transition blocks. Hence, for the BiDeep RNN, the GRU is replaced with a multi-layer deep transition GRU.

In this research, the final models (Sections 6.4 and 6.5) carry the BiDeep RNN architecture implemented with Marian, with four encoder layers (each with two transitional GRU cells) and four decoder layers (the first layer with four and the next layers with two transitional GRU cells). The same embedding and hidden state sizes are used as the deep transition model. The BiDeep models are equipped with tied embeddings, layer normalization, exponential smoothing, gradient clipping, and label smoothing (0.1). The model is optimized using Adam with the same parameters and the same stopping criterion as the deep transition model and was trained on two GPUs.

4.2 Transformer Model

The sequential nature of the recurrent encoder-decoder models with attention makes parallelization within training examples difficult, especially for long sentences. In addition, distant words may not affect each other's output without passing through many RNN steps or convolutional layers. To address these problems, Vaswani et al. [65] introduced self-attention. Their entirely attention-based new model introduced short paths between distant words and reduced the amount of sequential computation. The model architecture that they have introduced is called *Transformer*, a model that allows more parallelization, better translation quality, and less training time.

In this study, we use the Marian implementation of the Transformer models. Encoder and decoder depths are both set to six layers, employing eight-head multi-head attention. All Transformer models have been trained on four GPUs with early stopping if the word-level cross entropy does not improve after five 5,000 updates, up to 12 epochs. Different from the original model, the size of the position-wise feed-forward network has been set to 4096 instead of 2048 and the size of the embedding vector has been set to 1024 instead of 512, resembling Google's Transformer-Big

Table 4. Statistics of Augmented Corpora After Tokenization and Cleaning

		Synthetic					
Corpus	**Direction**	**Self-Trained**	**Back-Translated**	**Iteratively Back-Translated**	**Copied**	**Original**	**Total**
SETimes-clean	Both	–	–	–	–	207,373	207K
A	TR-EN	448,811	–	–	–	207,373	656K
B	TR-EN	1,994,892	–	–	–	207,373	2.2M
C	TR-EN	2,483,765	–	–	–	207,373 × 5	3.5M
D	TR-EN	–	2,404,835	–	–	207,373 × 5	3.4M
E	TR-EN	–	2,404,835	–	981,141	207,373 × 5	4.4M
F	TR-EN	2,483,765	2,404,835	–	981,141	207,373 × 5	6.9M
G	TR-EN	–	800,000	–	–	207,373	1M
H	TR-EN	–	–	2,399,595	–	207,373 × 5	3.4M
I	TR-EN	–	–	2,399,595	981,141	207,373 × 5	4.4M
J	EN-TR	–	2,486,627	–	–	207,373 × 5	3.5M
K	EN-TR	–	2,486,627	–	1,007,484	207,373 × 5	4.5M
L	EN-TR	–	800,000	–	–	207,373	1M
M	EN-TR	–	–	2,476,660	–	207,373 × 5	3.5M
N	EN-TR	–	–	2,476,660	1,007,484	207,373 × 5	4.5M

architecture. Although compromising from speed and memory usage, we observed improvement over the original model (Section 6.1).

In addition to dropout between Transformer layers (0.1), dropouts for Transformer attention (0.1) and Transformer filter (0.1) have been applied. As in the attentional encoder-decoder models, tied embeddings, layer normalization, exponential smoothing, gradient clipping, and label smoothing (0.1) have been adopted. To be compatible with the increase in the parameters, mini-batch size was fit into 8 GB of GPU memory. The best models according to the BLEU score for the validation set have been kept.

4.3 Data Augmentation

The low-resource setting of the Turkish-English pair (207K parallel sentences) has encouraged the use of monolingual corpora for augmenting the parallel data. Data augmentation is employed through self-training for the source-side and through copying and back-translation for the target-side. We use the WMT News Crawl 2020 dataset as the Turkish and English monolingual data.

We experiment with different types of data augmentation and different sizes to observe their effects on the NMT performance. Table 4 lists all of the corpora used in this work for both translation directions. SETimes-clean is the parallel corpus that all augmented corpora are based on as explained in Section 3.

Corpus A is used in the experiments where different model architectures and input variations are tested. In forming Corpus A, we trained a shallow Turkish-English attentional encoder-decoder model using the SETimes-clean corpus. The Moses scripts for tokenization, truecasing, and punctuation normalization [36] were applied to the parallel corpus. Joint BPE was employed for subword segmentation [57]. With the trained model, source-side (Turkish) monolingual data of 449K sentences were translated into English. The synthetic corpus obtained was combined with SETimes-clean, resulting in Corpus A with 656K sentences. To observe how the amount of self-trained data affects the translation quality, the same process was repeated with 4.5 times longer monolingual data to obtain Corpus B with 2.2M sentences.

After the results of different models and input variations have been obtained, the best input segmentation method was selected to be used in the final models. We built several corpora (corpora C through F and corpora J and K) with different combinations of self-trained data (by translating source-side monolingual data), copied data, and back-translated data (by translating target-side monolingual data) for a comprehensive evaluation of the final models. In the Turkish-English direction, the *Morph* input segmentation method (Section 4.4) was used on the SETimes-clean corpus, since ensemble models with this input variation yielded the best results in both test sets (see Table 11). An attentional encoder-decoder model with BiDeep architecture was trained on this corpus. The trained model was then used for self-training of 2.5M Turkish sentences. The self-trained parallel corpus obtained has undergone special cleaning steps, taking an example from Durgar El-Kahlout et al. [18]. A sentence pair was removed if the target sentence consists of only one word, the token count ratio between the target sentence and the source sentence is greater than 3, or a token in the target sentence repeats itself three times consecutively. After cleaning is complete, the synthetic corpus was paired with the SETimes-clean corpus. To prevent the ratio of synthetic over original from becoming too much in favor of the synthetic, the original parallel corpus was oversampled (copied) five times (shown as “x 5” in Table 4), forming Corpus C (3.5M sentences).

In addition to exploiting source-side monolingual data via self-training, we also incorporated target-side monolingual data via back-translation. For the Turkish-English direction, an English-Turkish BiDeep model was trained on the SETimes-clean corpus pre-processed with the *Morph* input segmentation method (Section 4.4). A total of 2.4M English sentences were back-translated using the trained model. The obtained synthetic back-translated corpus was cleaned in the same way as Corpus C and was combined with the five times oversampled SETimes-clean corpus, obtaining Corpus D (3.4M sentences).

The input variation *Morph* could not be used for the English-Turkish direction. Since the morphemes are different from the allomorphs and contain many phonetic variations, it is challenging to reconstruct a Turkish word from the morpheme-based input segmentations when the target language is Turkish. However, it is possible to desegment a Turkish sentence for the *Allomorph* input variation that yielded translation quality close to the best-performing segmentation method (*Morph*) in the Turkish-English direction. Therefore, the same back-translation approach was repeated for the English-Turkish direction by training a Turkish-English BiDeep model on the SETimes-clean corpus pre-processed with the *Allomorph* input segmentation method (Section 4.4), back-translating and cleaning 2.5M Turkish sentences, and combining with five times oversampled SETimes-clean to obtain Corpus J (3.5M sentences).

As the third type of data augmentation, we created a copied corpus of 1M sentences (Currey et al. [16]). A total of 1M English sentences for the Turkish-English direction and 1M Turkish sentences for the English-Turkish direction were taken from the monolingual corpora. For each, a bitext was formed with the source-side identical to the target-side. The English copied corpus was added to Corpus D to form Corpus E (4.4M sentences) and the Turkish copied corpus to Corpus J to form Corpus K (4.5M sentences). In addition, for Turkish-English translation, a corpus that contains all augmentations (self-trained, back-translated, copied corpora) was put together to form Corpus F (6.9M sentences).

Finally, we created corpora including iteratively back-translated data to see the effect of iterative back-translation in the Turkish-English NMT task. First, a BiDeep model was trained on the SETimes-clean corpus. Using this model, 800K sentences were back-translated and coupled with SETimes-clean, obtaining Corpus G and Corpus L. These corpora, each with 1M sentences, were used to train the second iteration model of the back-translation process. Afterward, 2.5M sentences were back-translated with these models and Corpora H and M were created by coupling with the

Table 5. Examples of BPE and WordPiece (BERT) Segmentation

	EN	TR
Original Sentence	There is no difference between those who covet them and those who watch them bursting with anger.	Gıpta eden ya da sinirden köpürerek izleyenler arasında fark yok.
BPE Segmented Sentence	There is no difference between those who cov@@ et them and those who watch them bur@@ sting with anger .	Gıpta eden ya da sinir@@ den kö@@ pür@@ erek izle@@ yenler arasında fark yok .
WordPiece Segmented Sentence	There is no difference between those who co ##vet them and those who watch them bursting with anger .	Gı ##pt ##a eden ya da sinir ##den köp ##ür ##erek izleyen ##ler arasında fark yok .

five times oversampled original parallel corpus. The final corpora we formed are Corpora I and N, which combine the iteratively back-translated and copied data.

4.4 Input Variations

Input representation is an important factor in translation quality, especially for low-resource settings. In addition to being low-resource, Turkish is also a morphologically rich language requiring special attention for word segmentation. In this work, mainstream word segmentation techniques and morphologically motivated segmentation techniques designed specifically for Turkish are used and compared in the scope of the Turkish-English NMT task.

Input segmentation methods are explained with examples and statistics in the following sections. In all scenarios, subword segmentation is applied after truecasing, punctuation normalization, and tokenization of the sentence. The tokenization process mentioned here is merely the separation of words and punctuation.

4.4.1 BPE and WordPiece. BPE is a word segmentation algorithm that encodes rare words via subword units [57]. The open vocabulary problem is tackled by creating a fixed-size vocabulary consisting of variable-length character sequences. In addition, translation of rare words, when represented with subword units, becomes easier to manage. The only hyperparameter of the BPE algorithm is the number of merge operations that determines the number of frequent character n-gram pairs that form a word or a subword when merged.

In this work, joint BPE is applied, which was observed by Sennrich et al. [57] to improve consistency between source and target segmentations and to be more effective with respect to learning BPE symbols separately. Joint BPE learning is achieved by the concatenation of source and target corpora and then applying the *subword-nmt* tool [49] on the concatenated corpus. The number of merge operations is set to 85,000. Segmented subwords carry the "@@" symbol at the end except for the rightmost subword of a word (see the example in Table 5).

The WordPiece algorithm is a word segmentation algorithm similar to BPE [70]. Once again, a provided number of merge rules are learned. Different from the BPE algorithm, which chooses the most frequent character n-gram pair, the pair that maximizes the language model likelihood is chosen.

For this subword tokenization scheme, the HuggingFace [68] implementation of BERT's [17] WordPiece tokenizers is used. For English, the case-sensitive *bert-base-cased* tokenizer [17] with a vocabulary size of 28,996, and for Turkish the *distilbert-base-turkish-cased* tokenizer [52] with a vocabulary size of 32,000, which is a distilled and lighter version of BERT [51], are used. After separately segmenting the Turkish and English sentences, subwords carry the "##" symbol at the beginning except for the leftmost subword of a word (see the example in Table 5).

Table 6. Input Variations of the Sentence *Gün geçtikçe bu tarz haberleri daha sık duyar hale geldik* (English Translation: *Day by day, we more frequently come to hear such news*)

Input Variation	**Segmented Sentence**
Morph	Gün geç _DHkçA bu tarz haber _lAr _SH daha sık duy _Ar hal _YA gel _DH _k .
ConcatMorph	Gün geç _DHkçA bu tarz haber _lArSH daha sık duy _Ar hal _YA gel _DHk .
LastMorph	Gün geç _DHkçA bu tarz haber _SH daha sık duy _Ar hal _YA gel _k .
Allomorph	Gün geç _tikçe bu tarz haber _ler _i daha sık duy _ar hal _e gel _di _k .
ConcatAllomorph	Gün geç _tikçe bu tarz haber _leri daha sık duy _ar hal _e gel _dik .
LastAllomorph	Gün geç _tikçe bu tarz haber _i daha sık duy _ar hal _e gel _k .
MorphTagsSuffix	Gün geç Adv_AsLongAs bu tarz haber A3pl P3sg daha sık duy Aor hal Dat gel Past A1pl .
MorphTagsAll	Gün Noun A3sg Pnon Nom geç Verb Pos Adv_AsLongAs bu Det tarz Noun A3sg Pnon Nom haber Noun A3pl P3sg Nom daha Adv sık Adj duy Verb Pos Aor A3sg hal Noun NoHats A3sg Pnon Dat gel Verb Pos Past A1pl . Punc
Multi-source	(1) Gün geç _tikçe bu tarz haber _ler _i daha sık duy _ar hal _e gel _di _k . (2) Gün geç Adv_AsLongAs bu tarz haber A3pl P3sg daha sık duy Aor hal Dat gel Past A1pl .

Although being quite similar to the BPE algorithm, BERT's tokenizer benefits from being pretrained on large amounts of data but has the drawback of using separate vocabularies for the two languages. Hence, it is intended in this study to make a comparison between the two methods. An alternative method for segmentation could be a multilingual language model such as XLM-RoBERTa, a Transformer-based masked language model trained on 100 languages by Conneau et al. [15]. The exploration of such models is left for future work.

4.4.2 Morphemes and Allomorphs. Morphemes are the smallest lexical items that carry a meaning. Allomorphs are different phonological variants of morphemes, where the difference can be in spelling or pronunciation. For instance, the ablative morpheme in Turkish is *DAn*, which has four allomorphs depending on the root word it is attached to: *tan, ten, dan, den.*[2] The complex morphology of the Turkish language led us to the idea of morphologically motivated input segmentations, meaning breaking up of a word into its morphemes via morphological analysis. The morphosyntactic and morphosemantic information carried by the morphemes and allomorphs is expected to be leveraged with several neural translation approaches. In this sense, comparison within these methods and with mainstream word segmentation methods (BPE, WordPiece) that are not linguistically motivated are carried out.

Before morphological processes, Turkish sentences are cleaned and truecased with the Moses scripts. Afterward, the Zemberek tokenizer [1] is used for the separation of words and punctuation. Table 6 shows the segmentation of an example sentence obtained with each of the methods explained in the following.

Morphological analysis of Turkish sentences is performed using the tool of Sak et al. [50]. After morphologically parsing the sentence, morphological disambiguation is applied on all possible parses of a word using the same tool and the best morphological analysis is selected (Appendix A.1). For the first input variation (*Morph*), the disambiguated morphological analysis of each word is used to extract its morphemes separated by a space. Each morpheme after the root morpheme

[2] In the convention used for Turkish morphology, uppercase letters in a morpheme indicate that the sound is phonologically conditioned depending on the previous morpheme. *A* changes into *a* or *e*, *D* changes into *d* or *t*, and *H* changes into *ı*, *i*, *u*, or *ü*. *S* is realized as *s* or drops, and *Y* is realized as *y* or drops.

starts with an underscore (“_”). Special extraction is used for capital letters from the original corpus since this knowledge is lost during the analysis.

The second input variation with this approach is obtained by concatenating the morphemes other than the root and is referred as *ConcatMorph*. In this case, the concatenated morpheme sequence carries an underscore at the beginning. Oflazer et al. [45] state that syntactic relation links between words are usually associated with the last morpheme or inflectional group of a word in Turkish. Based on this observation, we form another input variation (*LastMorph*) using only the root and the morpheme at the end of the word. This input segmentation method results in syntactic and semantic loss but decreases the amount of morphemes in a sentence, a rather interesting method to observe.

The use of morphemes in the input representation has the effect of vocabulary reduction since different phonetic variations of suffixes are represented with a single form. With the intention of observing if using finer morpheme categories improves translation quality, allomorphs are also investigated. The morphological analysis and disambiguation tool of Sak et al. [50] does not provide this functionality. Hence, the Zemberek tool [1] is used, which operates in a similar way as the tool of Sak et al. [50], but outputs allomorphs instead of morphemes (Appendix A.2). In parallel to the three morpheme segmentation methods, the input variations *Allomorph*, *ConcatAllomorph*, and *LastAllomorph* are built.

4.4.3 Morphological Tags. The use of morphological tags instead of morphemes has been previously employed in the Turkish-English MT task [2, 7]. Using the disambiguated analyses of the words by Sak et al. [50], two different segmentation methods are proposed in this work.

In the *MorphTagsSuffix* method, only the morphological tags that correspond to a suffix within the word are included in the input representation. The root word is represented in its surface form followed by the morphological tags of the morphemes. When a morpheme carries more than one morphological tag (e.g., -DHkçA[Adv+AsLongAs] for *geçtikçe* in Table 14 in Appendix A.1), the tags are combined with underscore. The *MorphTagsSuffix* setting is expected to produce a similar translation performance as *Morph* due to the fact that morphological tags have almost one-to-one correspondence with morphemes, as in the examples YH-[Acc], lAr-[A3pl], and NHn-[Gen].

In the second method (*MorphTagsAll*), all morphological tags and the type of the root tag are included in the input representation. This segmentation method significantly increases the number of tokens in a sentence and is thus expected to yield deteriorated results.

4.4.4 Multi-source. As mentioned previously, a large amount of Turkish-English parallel data is extremely difficult to come by. Data sparseness for this language pair makes it a necessity to acquire as much information from limited data as possible. Therefore, two different input segmentation methods can be exploited simultaneously hoping to capture semantic and syntactic properties from the morphemes as effectively as possible.

Pan et al. [46] trained a multi-source NMT model with a word-based encoder to capture word features and a knowledge-based encoder to capture linguistic features. Similar to this approach, we use two input variations, *Allomorph* and *MorphTagsSuffix*, together to train a single multi-source model. The former entails morphemes in their surface forms and the latter carries their morphological tags, thus clarifying the syntactic and semantic purpose of a morpheme inside the sentence. The intuition behind choosing *Allomorph* and *MorphTagsSuffix* together instead of other input variations that yield better translation performance such as *Morph* or *ConcatMorph* is that the usage of both morphological tags and morphemes in a multi-source setting resembles using the same information twice, since morphological tags has almost one-to-one correspondence to morphemes. Thus, we have chosen to incorporate allomorphs (which have phonetic and syntactic variations)

Table 7. Statistics of Input Variations on Corpus A

				+BPE		
Input Variation	**No. of Tokens**	**Vocabulary Size**	**Avg. Sentence Length**	**No. of Tokens**	**Vocabulary Size**	**Avg. Sentence Length**
Unsegmented	12,409,844	396,654	18.91	–	–	–
BPE	14,376,964	66,713	21.91	–	–	–
WordPiece	15,579,083	29,225	23.74	–	–	–
Morph	20,567,955	154,087	31.34	20,810,209	63,751	31.71
ConcatMorph	17,121,720	162,108	26.09	17,388,020	65,028	26.50
LastMorph	17,159,258	154,110	26.15	17,401,577	63,739	26.52
Allomorph	20,559,065	160,029	31.33	20,825,083	64,177	31.74
ConcatAllomorph	17,203,831	169,077	26.22	17,489,665	66,012	26.65
LastAllomorph	17,173,915	152,364	26.17	17,410,779	63,718	26.53
MorphTagsSuffix	20,570,495	148,191	31.35	20,791,690	63,163	31.69
MorphTagsAll	49,245,643	148,217	75.05	49,466,709	63,125	75.39

+BPE indicates further BPE segmentation after the morphologically motivated input variation is applied.

and morphological tags in the multi-source setting, seeing that they both yield high translation performance separately and can contribute to the translation task in different ways.

4.4.5 Statistics on Input Variations. The morphologically motivated segmentation methods explained previously break the words into smaller lexical items. However, rare words and proper nouns that cannot be recognized by the morphological analyzer are left unsegmented. To resolve such cases, the segmented input representation is further segmented via BPE. In this sense, for each linguistically motivated input variation, we employ both the original segmentation and the segmentation obtained by applying BPE to the original one. Further BPE segmentation has been applied to all morphologically motivated input variations, including the morphological tags.

Table 7 gives statistics for the input variations used in this work for Corpus A. For the morphological input variations, the left part of the table shows figures when the method is applied and the right part (indicated as +BPE) shows the corresponding figures with further BPE segmentation to observe the drop in vocabulary size (unique tokens) and the increase in average sentence length. Minor numerical differences between and within morphological variations (e.g., the difference between *Morph* and *Allomorph*) are due to the differences of morphological analyzers and/or due to some minor exceptions missed by the Python scripts that create each corpus.

We make the following observations on the corpus statistics with different input variations:

- All input variations increase the total amount of tokens in the corpus, all the while decreasing the vocabulary size.
- The smallest vocabulary size is obtained with WordPiece due to the fixed vocabulary size of the Turkish BERT tokenizer (32,000).
- Concatenation of morphemes/allomorphs and taking the last morpheme/allomorph decrease the average sentence length compared to using all morphemes/allomorphs separately.
- Applying BPE segmentation after morphologically motivated input segmentation reduces the vocabulary size by segmenting proper nouns, rare words, and long words that could not be segmented by the morphological analyzer.
- Using all morphological tags of a word (*MorphTagsAll*) more than doubles the average sentence length, thus lowering the expectation for high translation quality for this variation.

As for the English sentences, tokenization with the Moses script is applied when coupled with a morphologically segmented Turkish sentence. Further BPE segmentation is applied when coupled with a morphologically + BPE segmented Turkish sentence.

5 ENSEMBLE AND RESCORING

In this extensive study on Turkish-English NMT, systems with various model architectures, data augmentation methods, and input variations are trained and comparison between different settings is manifested. The reliability of a system is actually dependent on many factors, the initialization of parameters being one. Thus, to ensure that the translation performance of a system is reliable and taking into account the fact that exploiting multiple models improves translation quality [29], model ensembling is utilized during all experimentations in this work. Moreover, the observation of Liu et al. [37] of the imbalance in the quality of the output sentences (better translation quality of prefixes over suffixes) encourages the use of bidirectional decoding via rescoring [55].

Model ensembling in this work is carried out by training multiple models with different random initializations of model parameters. For all experiments except the final models, four L2R and four R2L models are trained, where each of the four models is randomly initialized with different seeds. For the final models, the number of models for each direction is decreased to 2 due to time and memory concerns on account of the largeness of training corpora.

After training is complete, during prediction, the test sentence is encoded and decoded by the L2R models and the output probabilities from the L2R decoders are averaged. The averaged word probabilities undergo beam search (beam size 50), and 50-best translation hypotheses are thus created. After n-best translation lists of the L2R models are originated, the 50 hypotheses of the test sentence are rescored with the R2L models by feeding the input sentence and the hypotheses to the R2L models. The hypothesis that obtains the highest score from the sum of L2R and R2L model scores is selected as the final translation.

6 EXPERIMENTS AND RESULTS

NMT of the Turkish-English language pair carries several difficulties, such as the sparsity of data, the rich morphology of Turkish, and the obvious dissimilarity of the two languages. Hence, choices like model architecture, amount of data, type of input representation, and hyperparameters significantly affect translation quality. The experiments carried out in this study are expected to enlighten the importance of these choices and to find an optimal solution to this difficult task. After observing various settings (Sections 6.1–6.3), the best model architectures, data augmentation methods, and input segmentation techniques are selected to train the final models (Sections 6.4 and 6.5). To ease in following the rest of this section, the workflow of all experiments conducted in this work is demonstrated in Figure 1.

Evaluation of the NMT models are carried out with the *mteval-v14.pl* Moses script and case-sensitive BLEU scores (BLEU-cased) for WMT17 (*newstest2017*) and WMT18 (*newstest2018*) test sets are reported and compared.

6.1 Model Architectures

Among neural architectures used in NMT, attentional encoder-decoder and Transformer architectures are the most widely adopted ones and both have yielded state-of-the art results in many scenarios and language pairs. The Transformer architecture has recently been more predominant. Capturing long-term dependencies via self-attention and allowing parallel computation of outputs have significantly improved translation quality. However, with regard to memory and time consumption, encoder-decoder models are much easier to train and are therefore still preferred and tried to be improved. In this work, we experiment with both attentional encoder-decoder and

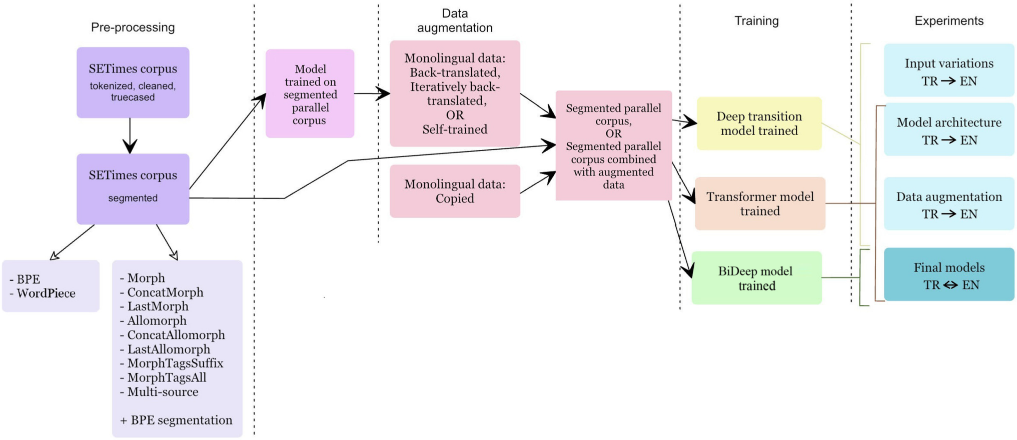
Pre-processing
Data augmentation
Training
Experiments
SETimes corpus
tokenized, cleaned, truecased
SETimes corpus
segmented
- BPE
- WordPiece
- Morph
- ConcatMorph
- LastMorph
- Allomorph
- ConcatAllomorph
- LastAllomorph
- MorphTagsSuffix
- MorphTagsAll
- Multi-source
+ BPE segmentation
Model trained on segmented parallel corpus
Monolingual data: Back-translated, Iteratively back-translated, OR Self-trained
Monolingual data: Copied
Segmented parallel corpus, OR Segmented parallel corpus combined with augmented data
Deep transition model trained
Transformer model trained
BiDeep model trained
Input variations TR → EN
Model architecture TR → EN
Data augmentation TR → EN
Final models TR ⇔ EN

Fig. 1. Workflow of the experiments.

Table 8. TR-EN News Translation (BLEU-Cased) Scores of Systems with Different Model Architectures

				newstest2017			newstest2018		
Model	**Input**	**No. of layers**	**Network Size**	**L2R Avg.**	**R2L Avg.**	**Ens.**	**L2R Avg.**	**R2L Avg.**	**Ens.**
Baseline	BPE	6, 6	512, 1024	–	–	15.12	–	–	15.64
Deep Transition	BPE	1(4), 1(8)	512, 1024	16.09	16.68	17.46	16.72	17.31	18.23
Deep Transition	WordPiece	1(4), 1(8)	512, 1024	16.23	16.46	17.63	16.82	17.08	18.26
Transformer	BPE	6, 6	512, 2048	14.35	13.82	15.50	14.59	14.09	15.72
Transformer	BPE	6, 6	1024, 4096	16.67	17.13	17.92	17.33	17.59	18.52
Transformer	WordPiece	6, 6	1024, 4096	**16.93**	**17.19**	**18.14**	**17.47**	**17.62**	**18.70**

Transformer models using different input representations and hyperparameters, so as to deduct the most suitable architecture for the low-resource Turkish-English language pair. The deep transition architecture is used as the encoder-decoder model.

All models are trained with Corpus A consisting of 207K original and 449K synthetic parallel sentences (a total of 656K). Separate systems are trained with BPE and WordPiece input representations. Table 8 shows the BLEU-cased score for each model and input representation. For each model architecture, in addition to the ensemble scores (shown as Ens.) of four L2R and four R2L models (Section 5), the average scores of the four L2R and the average scores of the four R2L models are also reported.

The shallow encoder-decoder model in the table denotes the model trained solely on the 207K SETimes-clean corpus and used for the translation of the 449K monolingual source data (Section 4.3). We use this model as baseline for comparison with the deep transition and Transformer models. As can be seen in the table, the systems trained on Corpus A outperform the baseline shallow NMT model by 1-3 BLEU, except for Transformer-BPE with network size (512, 2048), which shows very poor translation performance.

The positive effect of model ensembling can be observed for each system, where the BLEU score increases by up to 1.5 points. The Transformer architecture with network size (1024, 4096) yields better results than the deep transition encoder-decoder architecture for both input representations. An interesting point is that WordPiece input representation improves the L2R average of the systems compared to BPE, yet it deteriorates or very slightly improves the translation quality of the

Table 9. TR-EN News Translation (BLEU-Cased) Scores of Systems with Different Amounts of Data Augmentation

			newstest2017			newstest2018		
Model	**Input**	**Training Corpus**	**L2R Avg.**	**R2L Avg.**	**Ensemble**	**L2R Avg.**	**R2L Avg.**	**Ensemble**
Deep Transition	BPE	A (656K)	16.09	16.68	17.46	16.72	17.31	18.23
		B (2.2M)	16.60	17.40	17.81	17.24	18.08	18.61
Deep Transition	WordPiece	A (656K)	16.23	16.46	17.63	16.82	17.08	18.26
		B (2.2M)	16.14	17.20	17.09	17.07	17.68	17.90
Transformer	BPE	A (656K)	16.67	17.13	17.92	17.33	17.59	18.52
		B (2.2M)	**17.37**	**18.29**	**18.40**	**18.30**	**19.01**	**19.32**
Transformer	WordPiece	A (656K)	16.93	17.19	18.14	17.47	17.62	18.70
		B (2.2M)	16.98	17.98	17.78	17.87	18.56	18.76

R2L average. Furthermore, the Transformer architecture seems to be reacting better to WordPiece with respect to BPE than the encoder-decoder.

The improvement of WordPiece over BPE and of Transformer over the encoder-decoder is not too major for the 656K corpus at hand but is consistent over the ensemble results. The best BLEU-cased scores obtained from the model architecture experiments, 18.14 for WMT17 and 18.70 for WMT18, are for the ensemble Transformer architecture with network size (1024, 4096) and with WordPiece as the input segmentation method.

6.2 Data Augmentation

An analysis of previous work on the Turkish-English NMT task has shown that the available parallel corpora are far from being sufficient in size to obtain state-of-the-art results. For the purpose of eliminating the low-resource restriction, we form synthetic parallel data through data augmentation techniques and test its effect using different model architectures and input representations.

Transformer (with network sizes 1024 and 4096) and deep transition encoder-decoder models are trained as in Section 6.1. Each model architecture is experimented with BPE and WordPiece input representations. Two corpora are used to show the effect of data augmentation: Corpus A consisting of 207K original and 449K synthetic parallel sentences (total 656K sentences) and Corpus B consisting of 207K original and 2M synthetic parallel sentences (total 2.2M sentences). For each system, average BLEU-cased scores are presented in Table 9.

We note that the source-side monolingual data has been translated into English via a shallow NMT model with BPE as input representation. Thus, the lack of improvement for the Deep Transition-WordPiece and Transformer-WordPiece systems between corpora A and B can be related to the input representation of the self-training model. Deep Transition-BPE and Transformer-BPE systems, however, seem to consistently benefit from the increase in synthetic parallel data. When the amount of synthetic data is increased to 4.5 times its size, the Transformer-BPE system receives 0.48 and 0.80 higher BLEU scores for the WMT17 and WMT18 test sets, respectively.

6.3 Input Variations

The rich morphology of Turkish and the scarceness of data have led to the investigation of morphologically motivated input segmentation methods with respect to more general input representations like BPE and WordPiece. Linguistically motivated input representations proposed and compared in this work are Morphemes and Allomorphs (each used separately, in concatenated form, or by taking the last suffix), Morphological Tags (using only the tags that correspond

Table 10. L2R TR-EN News Translation (BLEU-Cased) Scores of Morphologically Motivated Input Segmentation Methods with and Without Further BPE Segmentation

Input	**Without BPE L2R**	**With BPE L2R Avg.**
Morph	16.29	17.21
ConcatMorph	16.28	17.28
LastMorph	15.08	16.03
Allomorph	16.19	17.19
ConcatAllomorph	15.87	17.06
LastAllomorph	14.98	16.11
MorphTagsSuffix	16.49	17.25
MorphTagsAll	14.91	15.49
Multi-source	16.43	17.32

to a suffix within the word or using all tags), and Multi-source (using both *Allomorph* and *MorphTagsSuffix*).

As stated in Section 4.4.5, morphological analyzers may not be able to segment all rare words or proper nouns into morphemes. Thus, after morphologically motivated input segmentation is applied to the tokens, the segmented input representation is further segmented via BPE. In the first experiment in this section, we test the effect of BPE segmentation on translation quality. For all input variations, we perform only one experiment for L2R models without further BPE segmentation, since we observed that this method is not promising. Obtaining more promising results from models with further BPE segmentation, we train four L2R models with this segmentation method to use them in ensemble. In Table 10, we report the BLEU-cased results for the WMT18 test set for one L2R model without further BPE segmentation and the average of four L2R models with further BPE segmentation. The BLEU scores show that linguistically motivated input decomposition methods work much better when coupled with BPE, observing an average of 0.94 BLEU improvement over nine input variations. Hence, from this point on, all mentioned morphologically motivated input decomposition methods are supported with further BPE segmentation.

The second experiment compares the performance of input representations using a deep transition attentional encoder-decoder model (Section 4.1.1) and model ensembling (Section 5). The systems are trained on Corpus A (656K sentences). The results are shown in Table 11. Among non-linguistically motivated methods, WordPiece representation performs slightly better than BPE. Comparison between linguistically and non-linguistically motivated methods shows that six of the linguistically motivated input representations improve translation quality over WordPiece and BPE, which is promising for the low-resource Turkish-English language pair.

For the analysis of the results of the linguistically motivated segmentation methods, we will refer to a Turkish word (*evdekilerle*, translated into English as “with the ones at home”) to aid in understanding of how each input representation looks. For each input variation, the corresponding segmentation is shown in the table.

The best input segmentation method is *Morph*, which improves the BLEU score by 0.96 and 0.60 points with respect to BPE for, respectively, *newstest2017* and *newstest2018* in the final ensemble results. Among the three variations of the morphemes approach, the best method is using all morphemes separately (*Morph*) instead of concatenating the morphemes (*ConcatMorph*) or using only the last morpheme (*LastMorph*).

Table 11. TR-EN News Translation (BLEU-Cased) Scores of Systems with Different Input Segmentation Methods

		newstest2017			**newstest2018**		
Input	**Example**	**L2R Avg.**	**R2L Avg.**	**Ensemble**	**L2R Avg.**	**R2L Avg.**	**Ensemble**
BPE	ev@@ dekilerle	16.09	16.68	17.46	16.72	17.31	18.23
WordPiece	evdeki ##lerle	16.23	16.46	17.63	16.82	17.08	18.26
Morph	ev _DA _ki _lAr _YlA	**16.71**	**17.35**	**18.42**	17.21	**17.78**	**18.83**
ConcatMorph	ev _DAkilArYlA	16.65	17.17	18.16	17.28	17.68	18.79
LastMorph	ev _YlA	15.62	16.03	17.14	16.03	16.39	17.39
Allomorph	ev _de _ki _ler _le	16.64	17.13	18.08	17.19	17.41	18.66
ConcatAllomorph	ev _dekilerle	16.48	16.89	18.11	17.06	17.58	18.65
LastAllomorph	ev _le	15.78	16.17	17.44	16.11	16.53	17.58
MorphTagsSuffix	ev Loc Adj-Rel A3pl Ins	16.52	17.09	18.24	17.25	17.71	18.58
MorphTagsAll	ev Noun A3Sg Pnon Loc Adj-Rel A3pl Pnon Ins	14.91	15.01	16.29	15.49	15.60	17.02
Multi-source	ev _de _ki _ler _le ev Loc Adj-Rel A3pl Ins	16.38	16.90	18.21	**17.32**	17.57	18.57

The *Allomorph* method yields a BLEU score in between BPE and *Morph*. This performance drop can be explained with the vocabulary reducing effect of *Morph* due to the elimination of phonetic variations of suffixes. In addition, usage of different morphological analyzers and disambiguators (Sak et al. [50] in morpheme-based methods and Zemberek [1] in allomorph-based methods) may also explain the difference, requiring further comparison on their performances. One exception is the *LastAllomorph* approach, which seems to outperform the *LastMorph* approach, yet it is far from competing with the non-linguistically motivated BPE or WordPiece especially in *newstest2018*. However, using only *LastAllomorph* may prove to be useful for the translation of very long Turkish sentences, where the syntactic and semantic loss could be compensated by the decrease in amount of tokens. The investigation of this is left for future work.

Even though there is an almost one-to-one correspondence between morphological tags and morphemes in the morphological analyzer of Sak et al. [50], the *MorphTagsSuffix* approach does not achieve the same translation performance as *Morph*. Being an approach adopted by researchers in the Turkish-English NMT task [2, 7], it is worth pointing out that using morphemes instead of their morphological tags turns out to be more successful. The *MorphTagsAll* method aims at incorporating different types of beneficial information produced by morphological analyzers, such as type of nouns, meaning, purpose, and person of morphemes. However, this approach results in unnecessarily long sentences, growing the average sentence length drastically. Thus, the BLEU score drops from 18.24 to 16.29 for *newstest2017* and from 18.58 to 17.02 for *newstest2018* with respect to *MorphTagsSuffix*.

Morphemes in their surface forms (allomorphs) may not present all of the syntactic and semantic information hidden within the morphemes. However, this information can be obtained from the corresponding morphological tags (*MorphTagsSuffix*). The idea behind the multi-source setting is to use *Allomorph* segmented input with a word-based encoder and *MorphTagsSuffix* with a knowledge-based encoder simultaneously, collecting as much information from a word as possible. However, the *Multi-source* input segmentation method improves the final ensemble BLEU score of *Allomorph* for *newstest2017* only by 0.13 and fails to improve the score for *newstest2018*. Using one of these input representations is more preferable with regard to memory and time consumption, thus showing that the combination of *Allomorph* and *MorphTagsSuffix* in a multi-source setting was not a good engineering choice for this task. Experimentation with other multi-source input variation pairs is left for future work.

A curious circumstance presents itself in the L2R and R2L averages of BLEU scores: for each input variation and for both test sets, the R2L models perform better than L2R. One of the causes for this observation may be the complex morphology of Turkish and the abundant usage of suffixes. This is similar to the argument of Liu et al. [37], where they observe better translation of the suffixes (last words) in the sentence with the use of R2L decoding, and we observe better translation of suffixes within a Turkish word in the source sentence.

Considering that scarce data makes it essential to represent the morphologically rich Turkish language as best as possible, determining the weaknesses and strengths of different input representations for the Turkish-English language pair is important. After comprehensive analysis, *Morph* and *MorphTagsSuffix* approaches combined with BPE are shown to be effective input segmentation methods, preferable to using only non-linguistically motivated methods like BPE and WordPiece.

6.4 Final Models (Turkish-English)

After comprehensive analysis on different model architectures, amounts of augmented data, and input variations, the final models are trained with the most optimal settings, aiming at high translation quality and generalization. With regard to model architecture, we observed that the Transformer architecture outperforms the attentional encoder-decoder model. This observation is expected to be confirmed in the final models. In the final experiments, we employ and compare the BiDeep model (Section 4.1.3) and the Transformer model (Section 4.2). As the input segmentation method, *Morph* followed by BPE segmentation is used, which gave the best results in input variations experiments. After observing that data augmentation increases the performance of the models in the previous experiments, all three data augmentation approaches are tested in the final models by using Corpus C (original and self-trained data), Corpus D (original and back-translated data), Corpus E (original, back-translated, and copied data), and Corpus F (original, self-trained, back-translated, and copied data).

Table 12 lists the results of the experiments conducted for determining the best models in the Turkish-English direction. For comparison with the most state-of-the-art result, the first row of the table gives the scores of the UEDIN system submitted to WMT18 [23]. The first part of the table shows the results obtained with the BiDeep and Transformer systems. In each of these, two L2R and two R2L models are trained and their averages are provided. The ensemble result is obtained by outputting the 50-best translation hypotheses by the two L2R models and then rescoring by the two R2L models.

The second part of the table corresponds to hybrid systems formed of BiDeep and Transformer models. In the first hybrid system, two L2R BiDeep models and two L2R Transformer models trained on Corpus D are combined. The four L2R models create the 50-best hypotheses, which are in turn rescored by the four R2L models of the two systems. In a similar fashion, the second hybrid system is formed of two L2R BiDeep models and two L2R Transformer models trained on Corpus D, and two L2R Transformer models trained on Corpus E. Rescoring is carried out by a total of six R2L models of these systems. For the third hybrid system, a total of six L2R Transformer models trained on corpora D, E, and F (two models for each corpus) are used to create the translation hypotheses. Rescoring is done by the six R2L models of these systems. The two L2R and two R2L BiDeep models trained on Corpus D are added to this third hybrid system to form the fourth hybrid system, comprised of eight L2R models trained on corpora D, E, and F, rescored by eight R2L models.

The third part of the table presents the iterative back-translation results (Section 4.3). Among the iterative back-translation models, the Transformer model trained on Corpus I yields the best

Table 12. TR-EN News Translation (BLEU-Cased) Scores of the Final Models

		newstest2017			newstest2018		
Model	**Training Corpus**	**L2R Avg.**	**R2L Avg.**	**Ensemble**	**L2R Avg.**	**R2L Avg.**	**Ensemble**
UEDIN (2018) [23]	*3.5M iteratively back-translated + copied*	–	–	*26.6*	–	–	*28.2*
BiDeep	C (3.5M) self-trained	17.98	18.41	18.95	18.26	18.51	19.10
Transformer	C (3.5M) self-trained	17.93	18.17	19.38	18.30	18.26	19.22
BiDeep	D (3.4M) back-translated	21.86	21.75	23.25	22.95	22.64	24.35
Transformer	D (3.4M) back-translated	22.22	**22.25**	24.04	24.00	**23.96**	25.75
Transformer	E (4.4M) back-translated + copied	**22.36**	21.38	23.95	**24.28**	22.83	25.61
Transformer	F (6.9M) back-translated + self-trained + copied	20.43	20.43	21.39	21.15	20.42	22.13
BiDeep + Transformer hybrid	D	22.04	22.00	24.37	23.47	23.30	26.21
BiDeep + Transformer hybrid	D, E	22.15	21.79	24.74	23.74	23.14	**26.38**
Transformer hybrid	D, E, F	21.67	21.35	24.58	23.14	22.40	25.86
BiDeep + Transformer hybrid	D, E, F	21.72	21.45	**24.79**	23.09	22.46	26.18
BiDeep	H (3.4M) iteratively back-translated	21.36	21.35	22.91	22.56	22.57	23.77
Transformer	H (3.4M) iteratively back-translated	22.20	21.90	23.70	23.70	23.44	25.39
Transformer	I (4.4M) iteratively back-translated + copied	22.02	22.16	23.74	23.90	23.57	25.39

ensemble BLEU scores. However, the results did not improve over the Transformer model trained on Corpus E, and they fall short in translation quality with respect to the hybrid models.

When we compare the self-training and back-translation strategies, we see that back-translation is much more effective than self-training on translation quality. With nearly the same amount of synthetic data (Corpus C and Corpus D), back-translation improves the BLEU score by

4.30 (*newstest2017*) and 5.25 (*newstest2018*) for the BiDeep model. We observe a similar effect for the Transformer model (4.66 increase for *newstest2017* and 6.53 increase for *newstest2018*). Addition of copied data (Corpus E) seems to improve the L2R models but degrades the R2L models, resulting in a similar translation performance with respect to using only back-translated data. When all data augmentation methods are used together (Corpus F), the performance seems to fall to a BLEU score between the self-trained and back-translated systems. Using back-translated and copied data instead of self-trained data seems a wiser choice for the Turkish-English NMT task, for fear of overgrowing the amount of synthetic data and decreasing generalization as in the case of Corpus F.

Ensembling multiple systems proves extremely rewarding regardless of the model architecture. A hybrid of the BiDeep and Transformer systems trained on Corpus D yields a higher BLEU than both systems. The best translation performance for the WMT17 test set is obtained from the hybrid of BiDeep and Transformer models trained on corpora D, E, and F (back-translated, self-trained, and copied data) with 24.79 BLEU. For the WMT18 test set, the best result is obtained from the hybrid of BiDeep and Transformer models trained on corpora D and E (back-translated and copied data) with 26.38 BLEU. Even though the L2R and R2L averages of the hybrid systems are not necessarily higher than those of the single systems, weaknesses of one system are compensated by the other's strength, pressing the importance of bidirectional decoding via model ensembling and rescoring.

Utilization of a morphologically motivated input segmentation method (*Morph*) shows its advantages in the given results, coming close to the state of the art by around 1.8 BLEU. Since we have focused on the linguistic motivation in this work, we could not perform extensive hyperparameter tuning, which may have aided to explain and decrease the difference between the results of the state-of-the-art model and our model. In addition, the choice of the specific portion of the monolingual data (WMT News Crawl) for back-translation and self-training may have affected the translation quality. Further experimentation on different amounts of back-translated data, deeper Transformer architectures, and more advanced ensembling methods are planned as future work.

6.5 Final Models (English-Turkish)

In this work, we mainly focus on observing the translation quality in the Turkish-English direction. The main reason for this is that some of the morphologically motivated input variations can only be applied if Turkish is in the source-side as mentioned in Section 4.3 due to the challenge of reconstructing a Turkish word from morpheme-based input variations when Turkish is the target language. Therefore, we train final models for the English-Turkish direction using the *Allomorph* segmentation method, which can be used to desegment a Turkish sentence and yielded close translation performance to the *Morph* segmentation method.

Similar to the final models in Section 6.4, BiDeep and Transformer models are used. For this direction, only back-translation, iterative back-translation, and copying are employed for data augmentation, seeing their dominance over self-training for the reverse direction. A Turkish-English BiDeep shallow NMT model is trained on the SETimes-clean corpus to form back-translated data from target-side monolingual data. The models are trained on Corpus J (original and back-translated data), Corpus K (original, back-translated, and copied data), Corpus M (original and iteratively back-translated data), and Corpus N (original, iteratively back-translated, and copied data). Ensembling is done similar to the Turkish-English direction with two L2R and two R2L models. Table 13 shows the results of the experiments and also the results of the UEDIN system submitted to WMT18 [23] for comparison.

When we look at the BiDeep and Transformer models trained on Corpus J and Corpus M, as in the reverse direction (Turkish-English), the results confirm that iterative back-translation does not

Table 13. EN-TR News Translation (BLEU-Cased) Scores of the Final Models

		newstest2017			**newstest2018**		
Model	**Training Corpus**	**L2R Avg.**	**R2L Avg.**	**Ensemble**	**L2R Avg.**	**R2L Avg.**	**Ensemble**
UEDIN (2018) [23]	*3.5M iteratively back-translated + copied*	–	–	*24.7*	–	–	*20.1*
BiDeep	J (3.5M) back-translated	17.67	17.28	18.89	14.98	14.35	16.08
Transformer	J (3.5M) back-translated	18.44	18.39	19.88	15.60	15.36	16.62
Transformer	K (4.5M) back-translated + copied	**18.90**	**19.11**	**20.19**	**16.04**	**16.00**	16.76
BiDeep	M (3.5M) iteratively back-translated	17.70	16.98	18.82	14.84	14.12	15.95
Transformer	M (3.5M) iteratively back-translated	18.28	18.07	19.82	15.37	15.06	16.43
Transformer	N (4.5M) iteratively back-translated + copied	18.80	18.71	19.71	15.96	15.78	**16.98**

yield better results with respect to regular back-translation. Comparison between the Transformer models trained on Corpus K and Corpus N also demonstrates better translation quality with back-translation with respect to iterative back-translation, with the exception of the ensemble score for the WMT18 test set. The additional computational time required for training different iterations of back-translation does not seem to pay off.

The best translation performance is obtained from the Transformer model trained on Corpus K for the WMT17 test set and Corpus N for the WMT18 test set. Hence, the advantage of employing two data augmentation techniques (back-translation and copying) where the amount of parallel data is increased to 4.5M sentences is apparent. As in the reverse direction, the benefits of model ensembling and rescoring can be observed, improving the BLEU score by 1 to 2 as compared to L2R and R2L averages. However, the English-Turkish translation scores are 3 to 4 BLEU scores lower than those of the state of the art. Thus, it is left for future work to analyze and enhance the NMT models for this direction, and perhaps to modify the iterative back-translation approach for both directions.

7 CONCLUSION

In this study, we approached the Turkish-English NMT task from a morphologically motivated angle, all the while incorporating state-of-the-art NMT architectures and data augmentation methods. Two architectures of the attentional encoder-decoder model, namely deep transition and BiDeep, have been trained and compared to the Transformer architecture. Scenarios that entailed different input representations and amounts of training data have led to the conclusion that the Transformer architecture, although costly in memory and time consumption, outperforms the attentional encoder-decoder models.

Parallel data has been augmented through three methods (self-training, back-translation, and copying). Experiments on data augmentation through self-training have shown that an increase in synthetic data results in better translation performance but is also dependent on the compatibility of input representations.

Encouraged by the rich morphology of Turkish, nine morphologically motivated input segmentation methods (based on Morphemes, Allomorphs, and Morphological Tags) and two non-morphologically motivated methods (BPE and WordPiece) have been experimented with and compared. Extensive experimentation has proven the success of morphologically motivated input segmentation for Turkish. Keeping all other parameters of the NMT models unchanged, the addition of linguistically motivated input segmentation on top of BPE has led to better translation quality for six of the proposed input representation methods. The best morphologically motivated input segmentation method has been selected to be *Morph*, outperforming BPE by 0.96 BLEU.

Final models have been trained with the BiDeep attentional encoder-decoder and Transformer architectures on augmented corpora of up to 6.9M sentences, with input in the form of *Morph* + BPE for the Turkish-English direction and *Allomorph* + BPE for the English-Turkish direction. The effectiveness of the morphologically motivated input scheme has been demonstrated with a BLEU score of 26.38 on the WMT18 test set from a Turkish-English BiDeep-Transformer hybrid system trained on back-translated and copied data. The importance of bidirectional decoding with ensemble and rescoring has been pressed, and the power of back-translation has been confirmed.

In future work, further experimentation with different amounts and ratios of original, back-translated, and copied data is planned. Iterative back-translation is planned to be further investigated. All of the proposed morphologically motivated input variations are expected to be incorporated in deep models to obtain better translation quality and to observe further benefits. Finally, contributions made to the Turkish-English NMT task are aimed to be extended to other language pairs containing Turkish.

APPENDIX

A MORPHOLOGICAL ANALYSIS AND DISAMBIGUATION

A.1 Morphemes

Morphological analysis and disambiguation of Turkish sentences has been performed using the tool of Sak et al. [50] to obtain the morphemes within a sentence. Table 14 shows the parse of the sentence used in Section 4.4. The Analysis column lists all morphological parses of a word and the Disambiguation column denotes the correct parse in this context.

Table 14. Morphological Analysis and Disambiguation (Sak et al. [50]) of the Sentence
Gün geçtikçe bu tarz haberleri daha sık duyar hale geldik

Word	Analysis	Disambiguation
Gün	(1) Gün[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] (2) gün[Noun]+[A3sg]+[Pnon]+[Nom]	(2)
geçtikçe	(1) geç[Verb]+[Pos]-DHk[Noun+PastPart]+[A3sg]+[Pnon]+CA[Equ] (2) geç[Verb]+[Pos]-DHkçA[Adv+AsLongAs]	(2)
bu	(1) bu[Pron]+[Demons]+[A3sg]+[Pnon]+[Nom] (2) bu[Adj] (3) bu[Det]	(3)
tarz	(1) tarz[Noun]+[A3sg]+[Pnon]+[Nom]	(1)
haberleri	(1) haber[Noun]+[A3sg]+lArH[P3pl]+[Nom] (2) haber[Noun]+lAr[A3pl]+[Pnon]+YH[Acc] (3) haber[Noun]+lAr[A3pl]+SH[P3sg]+[Nom] (4) haber[Noun]+lAr[A3pl]+SH[P3pl]+[Nom]	(3)
daha	(1) daha[Adv]	(1)
sık	(1) sık[Verb]+[Pos]+[Imp]+[A2sg] (2) sık[Adj] (3) sık[Adv]	(2)
duyar	(1) duy[Verb]+[Pos]+Ar[Aor]+[A3sg] (2) Duyar[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] (3) duyar[Adj]	(1)
hale	(1) hâl[Noun]+[NoHats]+[A3sg]+[Pnon]+YA[Dat] (2) hal(II)[Noun]+[A3sg]+[Pnon]+YA[Dat] (3) hale[Noun]+[A3sg]+[Pnon]+[Nom] (4) Hale[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]	(1)
geldik	(1) gel[Verb]+[Pos]+DH[Past]+k[A1pl] (2) gel[Verb]+[Pos]-DHk[Noun+PastPart]+[A3sg]+[Pnon]+[Nom] (3) gel[Verb]+[Pos]-DHk[Adj+PastPart]+[Pnon]	(1)
.	(1) . [Punc]	(1)

A.2 Allomorphs

Morphological analysis and disambiguation of Turkish sentences has been performed using the Zemberek tool [1] to obtain the allomorphs within a sentence. Table 15 shows the parse of the sentence used in Section 4.4. The Analysis column lists all morphological parses of a word and the Disambiguation column denotes the correct parse in this context.

Table 15. Morphological Analysis and Disambiguation (Zemberek [1]) of the Sentence *Gün geçtikçe bu tarz haberleri daha sık duyar hale geldik*

Word	Analysis	Disambiguation
Gün	(1) [gün:Noun,Time] gün:Noun+A3sg	(1)
geçtikçe	(1) [geçmek:Verb] geç:Verb|tikçe:AsLongAs→Adv	(1)
bu	(1) [bu:Det] bu:Det	(1)
tarz	(1) [tarz:Noun] tarz:Noun+A3sg	(1)
haberleri	(1) [haber:Noun] haber:Noun+A3sg+leri:P3pl (2) [haber:Noun] haber:Noun+ler:A3pl+i:Acc (3) [haber:Noun] haber:Noun+ler:A3pl+i:P3sg (4) [haber:Noun] haber:Noun+ler:A3pl+i:P3pl	(3)
daha	(1) [daha:Adv] daha:Adv (2) [daha:Noun,Time] daha:Noun+A3sg	(1)
sık	(1) [sık:Adj] sık:Adj (2) [sık:Adv] sık:Adv (3) [sıkmak:Verb] sık:Verb+Imp+A2sg	(1)
duyar	(1) [duyar:Adj] duyar:Adj (2) [duymak:Verb] duy:Verb+ar:Aor+A3sg (3) [duymak:Verb] duy:Verb|ar:AorPart→Adj	(2)
hale	(1) [hal:Noun] hal:Noun+A3sg+e:Dat (2) [hâl:Noun] hal:Noun+A3sg+e:Dat (3) [Hale:Noun,Prop] hale:Noun+A3sg (4) [hale:Noun] hale:Noun+A3sg	(1)
geldik	(1) [gelmek:Verb] gel:Verb+di:Past+k:A1pl (2) [gelmek:Verb] gel:Verb|dik:PastPart→Adj (3) [gelmek:Verb] gel:Verb|dik:PastPart→Noun+A3sg	(1)
.	(1) [.:Punc] .:Punc	(1)

ACKNOWLEDGMENTS

The numerical calculations reported in this article were fully performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

REFERENCES

[1] Ahmetaa. n.d. ahmetaa/zemberek-nlp. Retrieved April 1, 2021 from https://github.com/ahmetaa/zemberek-nlp.

[2] Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *Prague Bulletin of Mathematical Linguistics* 108 (June 2017), 331–342. https://doi.org/10.1515/pralin-2017-0031

[3] Jimmy Ba, J. Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv abs/1607.06450* (2016).

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'16): Conference Track Proceedings.*

[5] Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3028–3033. https://doi.org/10.18653/v1/D18-1338

[6] Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-Jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, et al. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the 5th Conference on Machine Translation.* 1–55.

[7] Emre Bektaş, Ertuğrul Yilmaz, Coşkun Mermer, and İlknur Durgar El-Kahlout. 2016. TÜBİTAK SMT system submission for WMT2016. In *Proceedings of the 1st Conference on Machine Translation (Volume 2: Shared Task Papers).* 246–251. https://doi.org/10.18653/v1/W16-2305

[8] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, et al. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the 2nd Conference on Machine Translation.* 169–214. https://doi.org/10.18653/v1/W17-4717

[9] Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the 3rd Conference on Machine Translation: Shared Task Papers*. 272–303. https://doi.org/10.18653/v1/W18-6401

[10] M. Asunción Castaño, Francisco Casacuberta, and Enrique Vidal. 1997. Machine translation using neural networks and finite-state models. In *Proceedings of the 7th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*. 160–167.

[11] Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the 4th Conference on Machine Translation (Volume 1: Research Papers)*. 53–63. https://doi.org/10.18653/v1/W19-5206

[12] Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 76–86. https://doi.org/10.18653/v1/P18-1008

[13] Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1965–1974. https://doi.org/10.18653/v1/P16-1185

[14] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of the 8th Workshop on Syntax, Semantics, and Structure in Statistical Translation (SSST-8)*. 103–111. https://doi.org/10.3115/v1/W14-4012

[15] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[16] Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the 2nd Conference on Machine Translation*. 148–156. https://doi.org/10.18653/v1/W17-4715

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805* (2018).

[18] İlknur Durgar El-Kahlout, Emre Bektaş, Naime Şeyma Erdem, and Hamza Kaya. 2019. Translating between morphologically rich languages: An Arabic-to-Turkish machine translation system. In *Proceedings of the 4th Arabic Natural Language Processing Workshop*. 158–166. https://doi.org/10.18653/v1/W19-4617

[19] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 489–500. https://doi.org/10.18653/v1/D18-1045

[20] Mikel L. Forcada and Ramón P. Ñeco. 1997. Recursive hetero-associative memories for translation. In *Biological and Artificial Computation: From Neuroscience to Technology*, José Mira, Roberto Moreno-Díaz, and Joan Cabestany (Eds.). Springer, Berlin, Germany, 453–462.

[21] Mercedes García-Martínez, Ozan Caglayan, Walid Aransa, Adrien Bardet, Fethi Bougares, and Loïc Barrault. 2017. LIUM machine translation systems for WMT17 news translation task. In *Proceedings of the 2nd Conference on Machine Translation*. 288–295. https://doi.org/10.18653/v1/W17-4726

[22] Jeremy Gwinnup, Timothy Anderson, Grant Erdmann, Katherine Young, Michaeel Kazi, Elizabeth Salesky, Brian Thompson, and Jonathan Taylor. 2017. The AFRL-MITLL WMT17 systems: Old, new, borrowed, BLEU. In *Proceedings of the 2nd Conference on Machine Translation*. 303–309. https://doi.org/10.18653/v1/W17-4728

[23] Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann, Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone, and Rico Sennrich. 2018. The University of Edinburgh's submissions to the WMT18 news translation task. In *Proceedings of the 3rd Conference on Machine Translation: Shared Task Papers*. 399–409. https://doi.org/10.18653/v1/W18-6412

[24] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*. 820–828.

[25] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *CoRR abs/1909.13788* (2019).

[26] Cong Duy Vu Hoang, Gholamreza Haffari, and Trevor Cohn. 2017. Towards decoding as continuous optimisation in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 146–156. https://doi.org/10.18653/v1/D17-1014

[27] Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. 18–24. https://doi.org/10.18653/v1/W18-2703

[28] Kenji Imamura, Atsushi Fujita, and Eiichiro Sumita. 2018. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. 55–63. https://doi.org/10.18653/v1/W18-2707

[29] Kenji Imamura and Eiichiro Sumita. 2017. Ensemble and reranking: Using multiple models in the NICT-2 neural machine translation system at WAT2017. In *Proceedings of the 4th Workshop on Asian Translation*. 127–134.

[30] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1–10. https://doi.org/10.3115/v1/P15-1001

[31] Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for WMT'15. In *Proceedings of the 10th Workshop on Statistical Machine Translation*. 134–140. https://doi.org/10.18653/v1/W15-3014

[32] Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael R. Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. *CoRR abs/2106.00941* (2021).

[33] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18): System Demonstrations*. 116–121.

[34] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1700–1709.

[35] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15): Conference Track Proceedings.*

[36] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. 177–180.

[37] Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 411–416. https://doi.org/10.18653/v1/N16-1046

[38] Gong-Xu Luo, Yating Yang, Rui Dong, Yan-Hong Chen, and Wenbo Zhang. 2020. A joint back-translation and transfer learning method for low-resource neural machine translation. *Mathematical Problems in Engineering* 2020 (2020), 1–11.

[39] Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1054–1063. https://doi.org/10.18653/v1/P16-1100

[40] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421. https://doi.org/10.18653/v1/D15-1166

[41] Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 11–19. https://doi.org/10.3115/v1/P15-1002

[42] Benjamin Marie, Rui Wang, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2018. NICT's neural and statistical machine translation systems for the WMT18 news translation task. In *Proceedings of the 3rd Conference on Machine Translation: Shared Task Papers*. 449–455. https://doi.org/10.18653/v1/W18-6419

[43] Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for neural machine translation. In *Proceedings of the 2nd Conference on Machine Translation*. 99–107. https://doi.org/10.18653/v1/W17-4710

[44] Kemal Oflazer. 1993. Two-level description of Turkish morphology. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics.*

[45] Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. *Building a Turkish Treebank*. Springer Netherlands, Dordrecht, 261–277.

[46] Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Multi-source neural model for machine translation of agglutinative language. *Future Internet* 12 (June 2020), 96. https://doi.org/10.3390/fi12060096

[47] Alberto Poncelas, D. Shterionov, A. Way, G. M. D. B. Wenniger, and P. Passban. 2018. Investigating backtranslation in neural machine translation. *arXiv abs/1804.06189* (2018).

[48] Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: (Volume 2: Short Papers)*. 157–163.

[49] Rsennrich. n.d. rsennrich/subword-nmt. Retrieved January 1, 2021 from https://github.com/rsennrich/subword-nmt.

[50] Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007. Morphological disambiguation of Turkish text with perceptron algorithm. In *Computational Linguistics and Intelligent Text Processing*, Alexander Gelbukh (Ed.). Springer, Berlin, Germany, 107–118.

[51] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *CoRR abs/1910.01108* (2019).

[52] Stefan Schweter. n.d. BERTurk: BERT Models for Turkish. Retrieved April 1, 2021 from https://github.com/stefan-it/turkish-bert.

[53] Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the 2nd Conference on Machine Translation*. 389–399. https://doi.org/10.18653/v1/W17-4739

[54] Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, et al. 2017. Nematus: A toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 65–68.

[55] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the 1st Conference on Machine Translation (Volume 2: Shared Task Papers)*. 371–376. https://doi.org/10.18653/v1/W16-2323

[56] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 86–96. https://doi.org/10.18653/v1/P16-1009

[57] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*. 1715–1725. https://doi.org/10.18653/v1/P16-1162

[58] Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 211–221. https://doi.org/10.18653/v1/P19-1021

[59] Yanyao Shen, Xu Tan, Di He, Tao Qin, and Tie-Yan Liu. 2018. Dense information flow for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 1294–1303. https://doi.org/10.18653/v1/N18-1117

[60] David R. So, Chen Liang, and Quoc V. Le. 2019. The evolved transformer. *CoRR abs/1901.11117* (2019).

[61] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (Volume 2) (NIPS'14)*. 3104–3112.

[62] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. *CoRR abs/1512.00567* (2015).

[63] Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*. 23–25.

[64] Francis M. Tyers and Murat Serdar Alperen. 2010. South-East European times: A parallel corpus of Balkan languages. In *Proceedings of the LREC Workshop on Exploitation of Multilingual Resources and Tools for Central and (South-) Eastern European Languages*.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. 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, Red Hook, NY, 1–11.

[66] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1810–1822. https://doi.org/10.18653/v1/P19-1176

[67] Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 791–802. https://doi.org/10.18653/v1/D19-1073

[68] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, et al. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *arXiv abs/1910.03771* (2019).

[69] Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19). 4207–4216. https://doi.org/10.18653/v1/D19-1430

[70] Yonghui Wu, M. Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv abs/1609.08144* (2016).

[71] Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1535–1545. https://doi.org/10.18653/v1/D16-1160

[72] Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. *CoRR abs/1801.05122* (2018).

[73] Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. 2020. Mirror-generative neural machine translation. In *Proceedings of the International Conference on Learning Representations*.

[74] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and W. Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics* 4 (2016), 371–383.

[75] Long Zhou, Jiajun Zhang, and Chengqing Zong. 2019. Synchronous bidirectional neural machine translation. *Transactions of the Association for Computational Linguistics* 7 (March 2019), 91–105. https://doi.org/10.1162/tacl_a_00256

[76] Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv abs/1503.03535* (2015).

Received 8 March 2022; revised 2 August 2022; accepted 6 November 2022