MORPHOLOGICALLY MOTIVATED INPUT VARIATIONS IN TURKISH-ENGLISH NEURAL MACHINE TRANSLATION

by

Zeynep Yirmibeşoğlu

B.S., Computer Engineering, Istanbul Technical University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2021

MORPHOLOGICALLY MOTIVATED INPUT VARIATIONS IN TURKISH-ENGLISH NEURAL MACHINE TRANSLATION

APPROVED BY:

Prof. Tunga Güngör ....
(Thesis Supervisor)

Assoc. Prof. Fatih Amasyalı ....

Assoc. Prof. Arzucan Özgür ....

DATE OF APPROVAL: 29.07.2021

ACKNOWLEDGEMENTS

I would like to thank Prof. Dr. Tunga Güngör for his kindness, patience, and guidance throughout my Master's study and research. I truly feel blessed to be under his advisory.

I am also extremely grateful to the jury members, Assoc. Prof. Fatih Amasyalı and Assoc. Prof. Arzucan Özgür, for their valuable insights, questions and contributions.

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

ABSTRACT

MORPHOLOGICALLY MOTIVATED INPUT VARIATIONS IN TURKISH-ENGLISH NEURAL MACHINE TRANSLATION

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. Tremendous 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 study, neural machine translation of the low-resource Turkish-English language pair is approached. State-of-the-art NMT architectures are employed and data augmentation methods that exploit monolingual corpora are used. The importance of input representation for the morphologically-rich Turkish language is pointed out, and a comprehensive analysis of linguistically and non-linguistically motivated input segmentation approaches has been made. Experiments on different input variations have proven the importance of morphologically motivated input segmentation for the Turkish language that carries a rich morphology. Moreover, superiority of the Transformer architecture over attentional encoder-decoder models has been shown for the Turkish-English language pair. Among the employed data augmentation approaches, back-translation has proven to be the most effective, and the benefit of increasing amount of parallel data on translation quality is confirmed. This thesis 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.

ÖZET

TÜRKÇE-İNGİLİZCE SİNİRSEL MAKİNE ÇEVİRİSİNDE MORFOLOJİK GÜDÜMLÜ GİRDİ VARYASYONLARI

Sinir ağlarının doğal dil işlemedeki başarısı, hızla makine çevirisinde ana yaklaşım haline gelen sinirsel makine çevirisinin (SMÇ) yolunu açmıştır. Kodlayıcı-kod çözücü (encoder-decoder) ağları, dikkat (attention) mekanizması ve Transformer mimarisi gibi atılımlarla çeviri performansında muazzam bir gelişme sağlanmıştır. Bununla birlikte, bir SMÇ sistemini eğitmek için büyük miktarda paralel verinin gerekmesi ve çeviri derlemlerinde kullanılan az rastlanmış kelimeler, henüz üstesinden gelinmemiş sorunlardır. Bu çalışmada, düşük kaynaklı Türkçe-İngilizce dil çiftinin sinirsel makine çevirisi ele alınmaktadır. Son teknoloji SMÇ mimarileri ve tek dilli derlemlerden yararlanılan veri artırma yöntemleri kullanılmıştır. Morfolojik açıdan zengin Türk dili için girdi temsilinin önemine dikkat çekilmiş ve dilbilimsel güdümlü ve dilbilimsel güdümlü olmayan girdi bölümleme yaklaşımlarının kapsamlı bir analizi yapılmıştır. Farklı girdi varyasyonları üzerinde yapılan deneyler, zengin bir morfoloji taşıyan Türkçe için morfolojik güdümlü girdi bölümlemenin önemini kanıtlamıştır. Ayrıca, Türkçe-İngilizce dil çifti için Transformer mimarisinin dikkat mekanizmasına sahip kodlayıcı-kod çözücü (attentional encoder-decoder) modellere göre üstünlüğü gösterilmiştir. Kullanılan veri artırma yaklaşımları arasında geri çevirinin en etkilisi olduğu kanıtlanmıştır ve paralel veri miktarındaki artışın çeviri kalitesine faydası doğrulanmıştır. Bu tez, farklı hiperparametrelerle eğitilen SMÇ mimarileri, veri büyütme yöntemleri ve girdi temsil teknikleri üzerine kapsamlı bir analiz sunmakta ve Türkçe-İngilizce SMÇ'nin düşük kaynak sorunu ile mücadele etmenin yollarını önermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT iv

ÖZET v

LIST OF FIGURES viii

LIST OF TABLES ix

LIST OF SYMBOLS xi

LIST OF ACRONYMS/ABBREVIATIONS xiii

1. INTRODUCTION 1

2. RELATED WORK 4

    2.1. Neural Machine Translation 4

    2.2. Low Resource Neural Machine Translation 7

    2.3. WMT17, WMT18 Tasks 10

    2.4. Input Variations 14

3. DATASET 18

4. METHODOLOGY 20

    4.1. Encoder-decoder Model 20

        4.1.1. Deep Transition Architecture 26

        4.1.2. Stacked Architecture 29

        4.1.3. BiDeep Architecture 31

    4.2. Transformer Model 32

        4.2.1. Scaled Dot-Product Attention 33

        4.2.2. Multi-Head Attention 34

    4.3. Data Augmentation 35

    4.4. Input variations 38

        4.4.1. Byte Pair Encoding 38

        4.4.2. WordPiece 39

        4.4.3. Morphemes and Allomorphs 40

            4.4.3.1. Morphemes 41

4.4.3.2. Allomorphs 42
4.4.4. Morphological Tags 44
4.4.4.1. Morphological Tags in Word 44
4.4.4.2. All Morphological Tags 44
4.4.5. Multi-source 44
4.5. Ensemble and Rescoring 47
5. EXPERIMENTS AND RESULTS 49
5.1. Model Architectures 49
5.2. Data Augmentation 51
5.3. Input Variations 53
5.4. Final Models 58
6. CONCLUSION AND FUTURE WORK 63
REFERENCES 65
APPENDIX A: COPYRIGHT LICENSE FOR FIGURES 4.3 AND 4.5 82
APPENDIX B: COPYRIGHT LICENSE FOR FIGURES 4.4 84

LIST OF FIGURES

Figure 4.1. Encoder-decoder architecture (self-drawn) 20

Figure 4.2. Attention mechanism, generating target word y_t [1] (self-drawn) . 23

Figure 4.3. Deep transition decoder [2] (License provided in Appendix A) . . . 27

Figure 4.4. Batch normalization versus layer normalization [3]. 28

Figure 4.5. Alternating stacked encoder [2] (License provided in Appendix A) 29

Figure 4.6. Transformer model architecture [4] (self-drawn) 33

Figure 4.7. Self-attention models [4] (self-drawn) 34

Figure A.1. Copyright license of Figures 4.3 and 4.5 82

Figure A.2. Creative Commons Attribution 4.0 International License 83

Figure B.1. Copyright license for the reuse of Figure 4.4, Page 1 85

Figure B.2. Copyright license for the reuse of Figure 4.4, Page 2 86

Figure B.3. Copyright license for the reuse of Figure 4.4, Page 3 87

Figure B.4. Copyright license for the reuse of Figure 4.4, Page 4 88

Figure B.5. Copyright license for the reuse of Figure 4.4, Page 5 89

LIST OF TABLES

Table 2.1. Turkish-English news translation results on the WMT17 test set . 11

Table 2.2. Turkish-English news translation results (official) on the WMT18 test set 13

Table 2.3. News translation BLEU scores of different input variations 16

Table 3.1. Corpus statistics 18

Table 4.1. Statistics of augmented corpora after tokenization and cleaning . . 36

Table 4.2. Examples of Byte Pair Encoding (BPE) and WordPiece (BERT) segmentation 39

Table 4.3. Morphological analysis and disambiguation (Sak et al. [5]) 42

Table 4.4. Morphological analysis and disambiguation (Zemberek [6]) 43

Table 4.5. Examples of input variations 45

Table 4.6. Statistics of input variations on Corpus A 47

Table 5.1. TR-EN news translation (BLEU-cased) scores of systems with different model architectures 51

Table 5.2. TR-EN news translation (BLEU-cased) scores of systems with different amounts of data augmentation 52

Table 5.3. Left-to-right TR-EN news translation (BLEU-cased) scores of morphologically motivated input segmentation methods with and without further BPE segmentation 54

Table 5.4. TR-EN news translation (BLEU-cased) scores of systems with different input segmentation methods 55

Table 5.5. TR-EN news translation (BLEU-cased) scores of the final models . 61

LIST OF SYMBOLS

a	Alignment model
c	Context vector
c_i	Context vector at time i
c_j	Context vector at time j
C	Context set *or* channel axis
d_k	Dimension of key
d_v	Dimension of value
D_s	Encoder stack depth
D_t	Decoder stack depth
e_{ij}	Associated energy for α_{ij}
g	Non-linear function
h_i	Annotation (encoder hidden state) at time i *or* forward state of i-th source word
h_j	Annotation (encoder hidden state) for input word j
H	Spatial axis
i	Position of output word *or* position of input word *or* timestep
j	Position of input word *or* position of output word *or* timestep
K	Key
L_s	Encoder recurrence depth
L_t	Decoder recurrence depth
N	Batch axis *or* source sentence length
p	Probability
Q	Query
s	Decoder hidden state
s_i	Decoder hidden state at time i
s'_j	Intermediate decoder hidden state at time j
s_t	Decoder hidden state at time t
t	Timestep

T	Number of timesteps
T_x	Number of input words
U_a	Model parameter
v_a	Model parameter
$v_{k,t}$	Intermediate vector for $DTGRU_k$ calculation
V	Value
w_i	Forward state of i-th source word
W	Spatial axis
W_a	Model parameter
W_{init}	Trained parameters of the decoder
W_K	Parameter matrix of key
W_O	Parameter matrix of output
W_Q	Parameter matrix of query
W_V	Parameter matrix of value
x	Input sentence
x_i	Word embedding of i-th source word
y	Output sentence
y_i	Word in output sentence at time i
y_j	Word in output sentence at time j
y_t	Word in output sentence at time t
α_{ij}	Weight of annotation for input word j and output word i
$\beta_{1,2}$	Parameters of Adam algorithm

LIST OF ACRONYMS/ABBREVIATIONS

1K	1 thousand
1M	1 million
ACL	Association for Computational Linguistics
ATT	Attention
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy Score
BPE	Byte-Pair Encoding
cGRU	Conditional Gated Recurrent Unit
DTGRU	Deep Transition Gated Recurrent Unit
EBMT	Example-Based Machine Translation
EMNLP	Empirical Methods in Natural Language Processing
EN	English
EOS	End of Sentence
GB	Gigabyte
GNMT	Google's Neural Machine Translation
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
IWLST	The International Workshop on Spoken Language Translation
L2R	Left-to-right
LM	Language Model
LSTM	Long Short-Term Memory
MCW	Morphologically Complex Word
MLM	Masked Language Model
MLP	Multilayer Perceptron
MT	Machine Translation
NAS	Neural Architecture Search
NLP	Natural Language Processing
NMT	Neural Machine Translation

NNLM	Neural Network Language Model
NSP	Next Sentence Prediction
OOV	Out-of-vocabulary
PBSMT	Phrase-based Statistical Machine Translation
POS	Part-of-speech
R2L	Right-to-left
RNMT	Recurrent Neural Network based Neural Machine Translation
RNN	Recurrent Neural Network
RNNLM	Recurrent Neural Network Language Model
SETimes	Southeast European Times
SMT	Statistical Machine Translation
SOS	Start of Sentence
TDA	Translation Data Augmentation
TR	Turkish
TRUBA	Turkish National e-Science e-Infrastructure
UEDIN	University of Edinburgh
WMT14	ACL 2014 9th Workshop on Statistical Machine Translation
WMT15	EMNLP 2015 10th Workshop on Statistical Machine Translation
WMT16	ACL 2016 1st Conference on Machine Translation
WMT17	EMNLP 2017 2nd Conference on Machine Translation
WMT18	EMNLP 2018 3rd Conference on Machine Translation

1. INTRODUCTION

Overcoming language barriers between people has been a concern of humankind for ages. Communication between people that speak different languages, availability of literary or professional text is achieved through human translation. However, accessing and maintaining human translation quality is, to this day, a costly and problematic issue. Unavailability and expensiveness of human translation, and advances in computer science and natural language processing (NLP) has led to the idea of automatic translation of languages: machine translation (MT).

The history of machine translation started with rule-based translation, relying on dictionaries and grammar rules. Along with the increase in computational power, data-driven, corpus-based approaches such as statictical machine translation (SMT) and example-based machine translation (EBMT) became more predominant. Extracting statistics from bilingual text has led to great success and state-of-the-art results, raising hope in quest of replacing human translation with MT.

Adoption and success of deep learning and neural networks in NLP has been the next big step in MT history, originating neural machine translation (NMT). Accommodating the entire machine translation system into an end-to-end neural network and eliminating excessive feature engineering, NMT gradually replaced SMT, becoming the new state-of-the-art, and the main technology behind commercial MT systems, such as Google [7] and Microsoft [8].

Incredible breakthroughs have been achieved in NMT with the introduction of the encoder-decoder network, attention mechanism, and the 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 discovered, taking NMT one step further.

In this research, Turkish-English neural machine translation has been investigated. Turkish-English NMT is an especially challenging task, due to the notable dissimilarity and low-resource setting of the Turkish-English language pair. Rich morphology of the Turkish language causes the extraction of information from unsegmented words to be rather troublesome.

In order to tackle this difficult task, a comprehensive analysis on state-of-the-art NMT model architectures, data augmentation techniques and input segmentation methods has been made. The attentional encoder-decoder model with deep transition and BiDeep architectures, and the Transformer architecture have been trained, pressing the importance of model and hyperparameter selection in Turkish-English NMT.

An exhaustive survey on the data sparsity issue in NMT has been carried out, resulting in a selection of three data augmentation approaches for this task: self-training, back-translation and copied data. These approaches have been exploited to expand the training corpus size from 207K sentences up to 6.9M sentences, observing the benefits of each approach separately, and together.

The most significant contribution of this study is aimed to be the implementation of 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. The advantages of employing linguistically motivated input representations in Turkish-English NMT are shown, in addition to an analysis of the strengths and weaknesses of each input variation.

The thesis is organized as follows: a comprehensive literature review on NMT is presented in Section 2. Statistics and resources of the datasets used in training of the NMT systems are given in Section 3. Model architectures, data augmentation and input segmentation methods are described in Section 4. Experimental results and their analysis are provided in Section 5. Finally, the work is summarized and future work is suggested in Section 6.

2. RELATED WORK

Among several machine translation approaches, including rule-based, statistical, example-based and neural machine translation (NMT), this research is based on the most recent methods, which lie around the neural machine translation approach. The history of neural machine translation is examined in Section 2.1.

Approaches to tackle the low-resource scenario in NMT, such as data augmentation (self-training, back-translation, copied data) and other semi-supervised methods are described in Section 2.2.

The Turkish-English news translation tasks in EMNLP's 2017 and 2018 Conferences on Machine Translation (WMT17, WMT18) pose a rich variety of models for Turkish-English NMT, pressing the importance of data augmentation in this low-resource setting. Turkish-English NMT systems in WMT17 and WMT18 are explained in detail in Section 2.3.

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. The most frequently used input representations, and linguistically inspired input variations are investigated in Section 2.4.

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 [9], where they introduced two feedforward neural networks called the encoder and the decoder, and Castaño et al. [10], where they compared subsequential transducers with neural networks in an MT task. Their works could not further be investigated, due to inadequate computational resources.

After the 2000s, it can be observed that neural networks have been used as an aiding tool in statistical machine translation. Bengio et al. [11] exploited feedforward neural language models for the target language. Zamora-Martinez et al. [12] have also used neural network language models (NNLMs) for source and target languages in their statistical machine translation (SMT) model. Neural networks have been used to learn the translation probabilities of phrase pairs [13], as reordering [14] and preordering models [15] and as joint models, augmenting the NNLM with a source context window [16]. These can be considered as examples of hybrid or joint models concerning neural networks [17].

End-to-end neural models that directly translate source sentence into target sentence are considered as pure examples of neural machine translation (NMT). In 2013, Kalchbrenner and Blunsom [18] introduced recurrent neural networks for translation modeling, laying the foundation of NMT. After this breakthrough, sequence to sequence NMT models started to be frequently 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 [19, 20].

The introduction of the encoder-decoder model is an important milestone in NMT, and was applied to English to French translation in the WMT14 (ACL 2014 9th Workshop on Statistical Machine Translation) translation task . This was one of the first neural machine translation models that outperformed baseline statistical machine translation models in such a large task. Afterwards, the encoder-decoder model was further enhanced with the addition of Bahdanau attention, and global/local attention mechanisms [1, 21], addressing the issue of translating long sentences.

Liu et al. [22] proposed a target-bidirectional model, trying to tackle the issue of unbalanced outputs in RNN (recurrent neural network) 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 is much higher than that of the suffixes. As a solution, they generate hypotheses from right-to-left (R2L) in addition to left-to-right (L2R), and enforce target agreement of these separate models via joint search. Bidirectional decoding has been further employed through rescoring n-best translation hypotheses [23], inference with linear relaxation [24], neural forward (for L2R models) and backward decoders (for R2L models) for asynchronous bidirectional inference [25], and a single bidirectional decoder for synchronous bidirectional inference [26]. Latest models prefer beam or greedy search for translation [25, 26].

The success of the attention mechanism brought with it the idea of self-attention, where attention is used not only between the encoder and decoders, but within them. Vaswani et al. [4] introduced two new self-attention mechanisms (Scaled Dot-Product and Multi-head attention), and a new architecture called the 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 [27]. Recent NMT architectures have also been incorporated in the realm of commercial MT systems, such as Google Translate [7] and Microsoft Translator [8].

Developing deep NMT models with better performance have got tremendous attention from researchers, resulting in advanced NMT models that are variants of vanilla Transformer and the attentional encoder-decoder. RNMT+ [28], an enhancement over Google's RNN-based GNMT (Google's Neural Machine Translation) model [7], consisted of 6 bi-directional LSTMs in the decoder, and took advantage of the Transformer model's multi-head additive attention. Bapna et al. [29] have trained a 16-layer Transformer model with a new attention mechanism: transparent attention. To better adjust gradient flow to depths of the encoder layers and optimize the gradient exploding/vanishing problem for deeper models, transparent attention creates weighted residual connections along the encoder depth. Wang et al. [30] train an even deeper Transformer with a 30-layer encoder, extending the work of Bapna et al. [29] with layer normalization and dynamic linear combination of layers. Searching for a simplified architecture with comparable performance, So et al. [31] apply neural architecture search (NAS), and train the original Transformer with 37.6% less parameters, outperforming it by 0.7 BLEU for the WMT14 English-German test set.

Employing state-of-the-art architectures, such as attentional encoder-decoders and Transformers, NMT toolkits for efficient and high speed training have been introduced, such as Nematus [32], OpenNMT [33], Tensor2Tensor [34], FairSeq [35] and Marian [36], a C++ re-implementation of Nematus. In this research, the Marian toolkit has been 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.

2.2. Low Resource Neural Machine Translation

Sparsity of sentence aligned parallel corpora significantly degrades the performance of NMT systems for low-resource language pairs. In order to tackle this issue, ways of extracting and exploiting the linguistic knowledge within monolingual corpora, which is 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. [37], where they came up with two methods of integrating recurrent neural network language models (RNNLM) trained on monolingual target-side data. Their first method was shallow fusion, where the translation hypotheses are rescored by the language model (LM). The second was deep fusion, where the hidden states of the LM are concatenated with those of the decoder. Another method presented itself in WMT15, through re-scoring n-best hypotheses of the NMT model with n-gram LMs [38].

Sennrich et al. [39] introduce two strategies to leverage monolingual data: empty (dummy) source sentences and synthetic source sentences. The former requires parallel examples with 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, 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 pseudo parallel data.

The back-translation approach has been further investigated, with comprehensive analysis on the amount of synthetic data, revealing an improvement of translation performance with larger amounts of back-translated data, until the point where the balance is too much in favor of the synthetic data [40]. Iterative back-translation also turned out to be beneficial for both low-resource and high-resource scenarios [41]. Introduction of back-translation into hierarchical transfer learning for low-resource Uygur-Chinese and Turkish-English language pairs has improved generalization with respect to baseline back-translation methods [42].

Enhancements over the original back-translation method have been made, by sampling multiple source sentences based on word distribution of output words [43], or sampling a single source sentence in addition to adding noise to beam search outputs [27], showing improvement in translation accuracy. Caswell et al. [44] revealed the role of noise in back-translation, which turned out to be helping the model distinguish between original and synthetic data. In turn, they extended this notion through a method called tagged back-translation, where synthetic data is explicitly labeled with the $< BT >$ tag, obtaining matching or higher scores on many different scenarios (low-resource, mid-resource, iterative). Another way of distinguishing between authentic and synthetic data to improve back-translation is through uncertainty-based confidence measures [45].

Improvement in translation quality that comes with data augmentation through original target-side monolingual data, has given birth to another strategy: copied monolingual data [46]. 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. Afterwards, the copied data is mixed with the original parallel corpus, to form the final training set.

Source-side monolingual data has also been seen as a source for data augmentation. Aiming to obtain better context representations, Zhang and Zong [47] flipped the back-translation approach, by translating source-side monolingual data into synthetic target data via self-learning. Their second approach is multi-task learning via two NMTs that simultaneously learn translation and source-sentence reordering. He et al. [48] more recently revisited self-training with injected noise, observing once again its smoothing effect. The work of Jiao et al. [49] asserts that self-training significantly improves 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. Wu et al. [50] adopt a strategy to leverage both sides, and observe that using both target and source sides improves translation quality with respect to only one of them. Other methods include an autoencoder that reconstructs the observed monolingual corpora [51], reinforcement learning with source and target-side LMs [52], iterative back-translation [41] and a mirror-generative NMT that can learn from the monolingual corpora by jointly training source-target, target-source NMT models and two language models [53].

Leveraging monolingual data in NMT has also been realized through pre-training, on the grounds of its effectiveness in language modeling and language understanding for many NLP tasks (Named Entity Recognition, Question Answering, etc.). Among the most widely used pre-training models, Embeddings from Language Models (ELMo) [54] is a deep bidirectional language model, pre-trained on large-scale unlabeled data. Bidirectional Encoder Representations from Transformers (BERT) [55] is a Transformer model, pre-trained with the masked language model (MLM) and next sentence prediction (NSP) objectives. After pre-training, the ELMo and BERT models can be fine-tuned for various tasks.

An example of integrating pre-training to NMT is by feeding ELMo word embeddings as input to the encoder or decoder of the Transformer [56]. Song et al. [57] have pre-trained a MAsked Sequence to Sequence (MASS) model with the objective of reconstructing a sentence with missing (masked) parts. Afterwards fine-tuning their model, they improved the state-of-the-art results of English-French translation, with 37.5 BLEU. Similarly, a denoising autoencoder that pre-trains sequence-to-sequence models with the object of reconstructing a corrupted text, BART [58] contributes to Romanian-English MT with an increase of 1.1 BLEU. A multilingual application of BART, mBART [59], reaffirms the success of pre-training in supervised and unsupervised MT for sentence and document levels.

Another method to alleviate the negative effects of the low-resource scenario is translation data augmentation (TDA) [60], inspired by data augmentation techniques in computer vision. Existing parallel sentences, especially ones with low-frequency words are altered with the help of long short-term memory (LSTM) language models, by substituting rare words with more common words, at the same time keeping the sentence plausible.

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 is extremely important performance boosters [61]. The domination of NMT over phrase-based statistical machine translation (PBSMT) in the low-resource setting for far less parallel training data has also been confirmed.

2.3. WMT17, 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 [62]. 7 systems have been submitted for the Turkish-English direction.

In this study, the three Turkish-English neural machine translation systems in WMT17 and their performances are taken into account (Table 2.1). All reported BLEU scores are of official submissions in WMT17, except for UEDIN's improved result in 2018 for the WMT17 test set.

The low-resource characteristic of the Turkish-English language pair (approximately 220,000 parallel sentences in the SETimes corpus) and the need for exploiting monolingual data due to its availability, all NMT systems with submissions in Turkish-English have used back-translation (automatic translation of target data into source data), approaching this technique in different ways.

Table 2.1. Turkish-English news translation results on the WMT17 test set.

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

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) [63]. The back-translated data amount was kept at around 150K sentences to abide with original-to-synthetic ratio. They obtained a 17.91 cased BLEU score from an ensemble of two Turkish-English models trained with a dropout of 0.3, and two models with a dropout of 0.2. 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 of back-translated data amount is significantly beneficial, seeing the 4.6 BLEU score improvement with 1M sentences as opposed to 150K sentences where the original-to-synthetic ratio is preserved.

The AFRL-MITLL system in WMT17 employed an iterative approach for back-translation [64], where they first created a Turkish-English statistical machine translation model with Moses, creating the first back-translated batch of data (around 5 million sentences). Afterwards they trained an English-Turkish Marian system on the parallel data and the back-translated data from the Moses model, and decoded the English monolingual data (around 9 million sentences). Finally, two left-to-right (L2R) Marian models and one right-to-left (R2L) Nematus model was trained on the parallel data and the back-translated data from the previous Marian model. The two L2R Marian models ensemble decoded, and the R2L Nematus model rescored the n-best lists to produce the final translation output. They combined their Turkish-English ensemble system, with their OpenNMT system where they used iterative back-translation on 800K sentences, and the aforementioned phrase-based Moses system, and submitted their result for newstest2017 in cased BLEU as 18.05.

University of Edinburgh's (UEDIN) system in WMT17 [65] used a stacked attentional encoder-decoder architecture proposed by [66] where the LSTM layers are stacked, and residual connections are used between stack layers. They trained shallow NMT models to back-translate 400K sentences. They also copied the monolingual corpus, and converted it into bitext, where source and target sides were identical. Final training corpus consisted of parallel, copied and back-translated data with 1:2:2 ratio. An ensemble of four left-to-right Nematus models was used for obtaining the 50 best translation hypotheses, which were in turn rescored by an ensemble of four Nematus right-to-left 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 [67]. Results of official submissions of the 2 systems for the Turkish-English direction are given in Table 2.2.

Table 2.2. Turkish-English news translation results (official) on the WMT18 test set.

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

The NICT system in WMT18 [68] incrementally trained their Marian Transformer models, increasing the amount of their back-translated data at each iteration. They first trained a Turkish-English and English-Turkish NMT system with the parallel data (approximately 220,000 parallel sentences in the SETimes corpus), and back-translated 200K sentences with each of these models. Afterwards, the two sets of synthetic parallel data are mixed with the original parallel corpus, to generate the next NMT models. This operation is performed 4 times, where the amount of back-translated data is doubled at each iteration, finally reaching 1.6M 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.

University of Edinburgh's (UEDIN) system in WMT18 [69] 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 [2], used with multi-head and multi-hop attention. Multi-head attention entails an MLP attention mechanism with a tanh hidden layer before a soft-max layer for the attention heads. Multi-hop attention includes attention hops introduced between the deep transition GRU layers in the decoder. Using the deep RNN setting, a back-translation system is trained using only the 200K parallel corpus. Using this model, 800K sentences are back-translated, creating a second back-translation system with the combination of the parallel corpus and the synthetic corpus (1M sentences). Afterwards, 2.5M sentences are back-translated with the second deep RNN model. For the final Marian Transformer models, one setting of the training corpus is the 2.5M synthetic sentences, and an addition of the parallel corpus oversampled 5 times (1M sentences). The second setting is the previous setting, with the addition of 1M copied data, obtained the same way as in the WMT17 task. 6 independently trained left-to-right models are used for translation, and 3 right-to-left for rescoring, yielding an official 26.9 BLEU score for the *newstest2018* test set (Table 2.2). Their best system received 28.2 BLEU (for *newstest2018*) after the shared task submission, reported as state-of-the-art in Section 5.4. They also improved their state-of-the-art submission for the WMT17 shared task, obtaining 26.6 BLEU for the *newstest2017* test set (Table 2.1).

All aforementioned models in the WMT17 and WMT18 tasks used byte pair encoding (BPE) [70] as the input scheme, with the *subword-nmt* tool [71].

2.4. Input Variations

Large vocabularies and out-of-vocabulary (OOV) words have been the focus of researchers due to the open vocabulary setting of NMT. To cope with the increase of training complexity due to large target vocabularies, Jean et al. [72] proposed importance sampling, exploiting a small subset of the vocabulary. Other techniques include a post-processing step that look up OOV words from a dictionary [73], and the representation of only OOV words as character embeddings [74]. Addressing both the OOV and the morphologically complex word (MCW) problem, Sennrich et al. [70] proposed their own word segmentation scheme, called *byte pair encoding* (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 MCW 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 an incredible 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, expecting better translation quality if the correct segmentation of morphemes inside a Turkish word is achieved.

Gülçehre et al. [37] employed an encoder-decoder model with Bahdanau attention, leveraging monolingual data via shallow and deep fusion. Regarding the input of their NMT model, they tokenized the Turkish sentences using Zemberek [6], followed by morphological analysis and disambiguation using Sak et al.'s [75] tool, afterwards removing non-surface morphemes (part-of-speech tags, etc.). Their NMT system, supported with deep fusion LM, reached a 20.56 BLEU for The International Workshop on Spoken Language Translation 2014 (IWLST14) test set. The same pre-processing approach was employed by Shen et al. [76] in their densely connected NMT system, obtaining 24.54 BLEU score on the IWLST14 test set.

Sennrich et al. [39] relied on the same architecture and the same pre-processing for Turkish sentences as Gülçehre et al. [37], differing in their usage of the monolingual data. Asserting that an encoder-decoder network can already model the probability distribution of a target word given its previous target words, they use back-translation and dummy source sentences (empty source side) instead of language models, to incorporate monolingual data. Their single model received a 20.4 BLEU for the IWLST14 test set.

Bektaş et al. [77] tokenized the Turkish sentences using the Moses tokenizer, followed by Oflazer's [78] morphological analyzer, and Sak et al.'s [75] morphological disambiguator 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. [79] also followed the same pre-processing approach, but included the root and all suffix tags in the Turkish input representation of their NMT model.

Table 2.3. News translation BLEU scores of different input variations.

					Test set	
System	**Model**	**Input**	**Parallel Data**	**Monolingual Data**	**IWLST14**	**WMT16**
Gülçehre et al. [37]	Attentional encoder-decoder + RNNLM	Morph. analysis + disamb.	160K	Not specified	20.56	-
Shen et al. [76]	Densely connected attentional encoder-decoder	Morph. analysis + disamb.	360K	-	24.54	-
Sennrich et al. [39]	Attentional encoder-decoder	Morph. analysis + disamb.	320K	3.2M back-translated	20.4	-
Bektaş et al. [77]	(Hierarchical) Phrase-based SMT	Morph. analysis + disamb. (Only morphemes in word)	208K	28M	-	16.01
Pan et al. [80]	Transformer	Multi-source (Lemma + POS + Morph. tag)	355K	-	27.37	-

Pan et al. [80] 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 (POS) tag, and the morphological tag. They used BPE for segmentation, followed by the Zemberek tool, and Sak et al.'s [75] morphological disambiguator . Their multi-source model achieved 27.37 BLEU score for the IWLST14 test set.

For better comparison, the aforementioned systems are summarized in Table 2.3.

3. DATASET

The SETimes (Southeast European Times) corpus is a parallel corpus gathered from news articles in 10 Balkan languages, containing 45 bitexts [81, 82]. The Turkish-English SETimes parallel corpus, consisting of 207K sentences, has been used in this research, where sentences have been tokenized and cleaned (sentences with less than 1 and more than 80 tokens) using the Moses cleaning scripts [83] before truecasing and further word segmentation. The SETimes-clean corpus has been used for training. Corpus statistics can be seen in Table 3.1.

Table 3.1. 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.	2,494,930	40,701,743	863,004	-	-
WMT News Crawl (EN)	Aug.	3,409,247	-	-	92,807,980	591,787

For monolingual data, the WMT News Crawl 2020 dataset [84] has been utilized. 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. Only the used portions of the corpora are reported in Table 3.1, usage denoted as “Aug.” for data augmentation.

For all models, the WMT16 test set (*newstest2016*) has been used for validation (development). WMT17 (*newstest2017*) and WMT18 (*newstest2018*) test sets have been used for testing.

4. METHODOLOGY

4.1. Encoder-decoder Model

The encoder-decoder architecture (Fig. 4.1) can be considered a dominating architecture in neural machine translation, where recurrent neural networks (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.

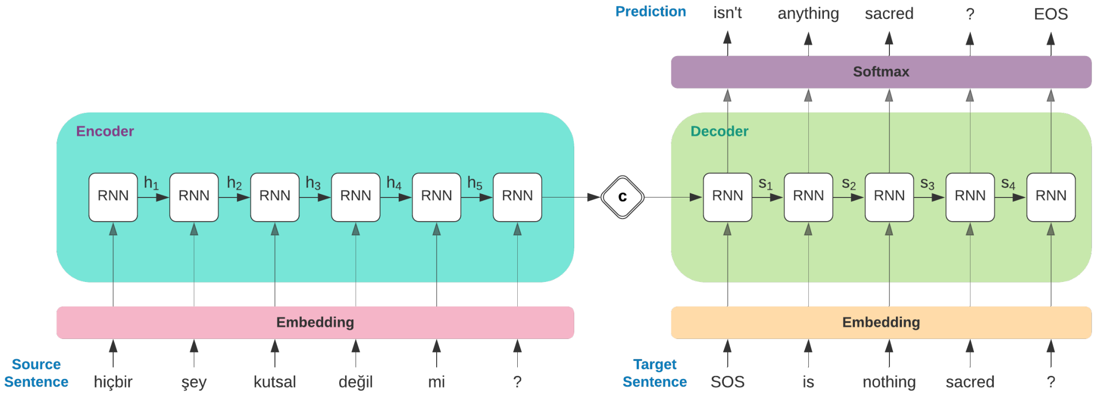


Figure 4.1. Encoder-decoder architecture (self-drawn)

The encoder consists of LSTM/GRU cell(s). It takes the input sequence, extracts the information and stores it in its internal states. Thus, the input sequence is reduced into a context vector. The output of the encoder is disregarded. The decoder usually follows a similar architecture to the encoder (LSTM/GRU). The states of the decoder are initialized to the final states of the encoder (the context vector). Thus, the decoder can generate the translated sequence based on the encapsulated information from the encoder. A softmax activation function is applied to the last layer of the decoder, to introduce non-linearity to the network.

During training, the decoder starts from the first token in the input sequence (SOS - start of sentence token in Fig. 4.1), and learns to predict the next word, until the end of the sentence. During testing (inference/translation), the decoder is given the start of sentence token (SOS), and predicts the entire sequence, word by word, until the end of sentence token (EOS) is reached.

If we formulate the translation process of the decoder, denoting the output sentence with y, we can state that the decoder is trained to predict the next word y_t given the context vector c and all previously predicted words $(y_1, y_2 ... y_{t-1})$. Hence, the probability over the final translation y becomes:

$$p(y) = \prod_{t=1}^{T} p(y_t | \{y_1, ..., y_{t-1}\}, c) \tag{4.1}$$

Each conditional probability is the output of a non-linear function g that returns the probability of y_t given the previous word y_{t-1}, the decoder hidden state s_t and the context vector c:

$$p(y_t | \{y_1, ..., y_{t-1}\}, c) = g(y_{t-1}, s_t, c) \tag{4.2}$$

The most significant handicap of the encoder-decoder network in NMT is poor translation performance for long sentences. Feeding the input sentence to the network in reverse order may aid in resolving short-term dependencies in the dataset [20]. However, the problem of long-term dependencies, and preserving the integrity of the target sentence for a long input sentence turned out to be an important issue to be resolved. Representing the information within a sentence in a fixed-length vector may not be adequate to encode a long sentence with a complicated structure [19]. To solve this issue, the concept of attention has been introduced.

Bahdanau et al. [1] introduced a new model, where the constant context vector $c(x)$ that represented the whole input sentence, is replaced by a series of context vectors $c_j(x)$ for each time step j. Thus, the attentional decoder can focus on only a part of the sentence that is most relevant/important when generating the next word: aligning and translating at the same time.

Transforming Eq. 4.2 with the attentional context vector c_i, the conditional probabilities become:

$$p(y_i|y_1, ..., y_{i-1}, x) = g(y_{i-1}, s_i, c_i), \tag{4.3}$$

where the hidden state s_i for time i is given by the formula:

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

The difference from the encoder-decoder architecture is that, the probability of a target word y_i is conditioned on a distinct c_i , which is a weighted sum of a series of annotations $(h_1, ..., h_{T_x})$ that contain information about the whole input sequence, with a strong focus on the words neighbouring the i-th word [1].

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \ ,$$

where the weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})}$$

$$e_{ij} = a(s_{i-1}, h_j)$$

a is an alignment model that scores the match between the inputs around position j and the output at position i based on the previous decoder hidden state s_{i-1} and the j-th annotation (encoder hidden state) h_j. The alignment model can be jointly trained as a feedforward neural network along with the encoder-decoder components. The system proposed by Bahdanau et al. [1] can be seen in Fig. 4.2.

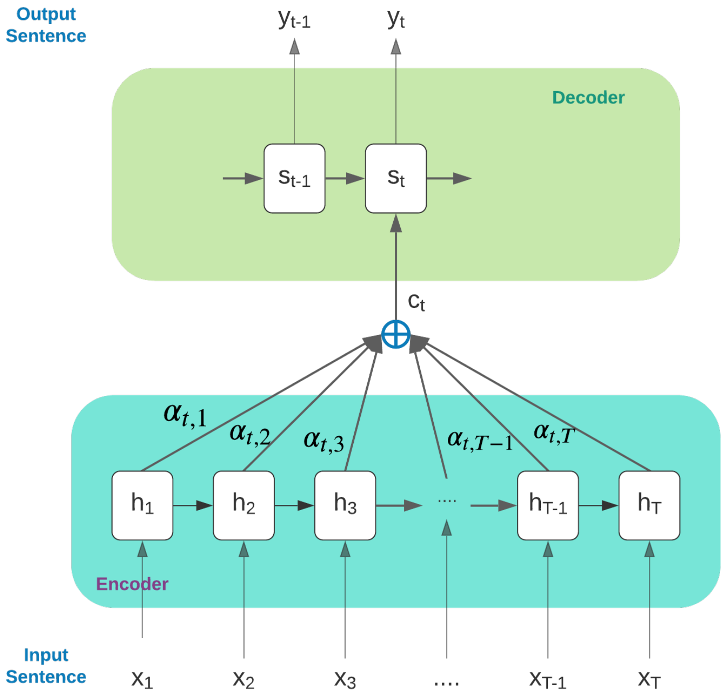


Figure 4.2. Attention mechanism, generating target word y_t [1] (self-drawn)

In this study, the Marian [36] implementation of the encoder-decoder model with Bahdanau attention has been used. Marian's attentional encoder-decoder is equivalent to that of Nematus [32], which follows the architecture proposed by Bahdanau et al. [1], with the following differences:

- As opposed to initializing the decoder hidden state with the last annotation in the backward encoder state

  $$s_0 = tanh(W_{init}\overleftarrow{h_1}) \tag{4.4}$$

  the decoder hidden state is initialized with the average of the source annotation:

  $$s_0 = tanh\left(W_{init}\frac{\sum_{i=1}^{T_x} h_i}{T_x}\right) \tag{4.5}$$

- A novel conditional GRU with attention, $cGRU_{att}$, is implemented, where the previous hidden state s_{j-1}, the entire context set $C = \{h_1, ..., h_{T_x}\}$, and the previously translated symbol y_{j-1} are used to update the hidden state s_j at position j, to be used in the prediction of symbol y_j, shown as $s_j = cGRU_{att}(s_{j-1}, y_{j-1}, C)$.

  $cGRU_{att}$ is made up of two GRU state transition blocks and an attention mechanism between them. The first GRU combines the previously translated symbol y_{j-1} and the previous hidden state s_{j-1}, generating an intermediate representation $s_j^{'}$, shown as $s_j^{'} = GRU_1(s_{j-1}, y_{j-1})$.

  The attention mechanism between the GRUs takes in the entire context set C and the intermediate hidden state $s_j^{'}$, outputting the context vector c_j:

$$\begin{aligned}
\mathbf{c}_j &= ATT(C, s_j^{'}) = \sum_{i}^{T_x} \alpha_{ij} h_i, \\
\alpha_{ij} &= \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{kj})}, \\
e_{ij} &= v_a^T tanh(U_a s_j^{'} + W_a h_i),
\end{aligned} \tag{4.6}$$

where α_{ij} is the normalized aligment weight between the ith source symbol and jth target symbol, and v_a , U_a and W_a are model parameters. In turn, the second GRU creates the hidden state s_j of the $cGRU_{att}$ with the help of $s_j^{'}$ and the context vector c_j, shown as $s_j = GRU_2(s_j^{'}, c_j)$.

The combination of RNN blocks occurs recurrently at the level of the entire cGRU layer, instead of indiviual recurrence in the GRU blocks, resembling *deep transition* RNNs [85].

- *tanh* non-linearity is introduced to the feedforward hidden layer of the decoder, instead of maxout before the softmax layer.
- Additional biases are not used in the encoder and decoder word embedding layers.
- Decoder implementation is simplified by employing *Look, Update, Generate* decoder phases, rather than *Look, Generate, Update* in Bahdanau et al. [1].
- Multi-source encoder-decoder networks can be trained. This allows the exploitation of multiple linguistic features, in that the final embedding is the concatenation of each feature embedding [86]. Junczys-Dowmunt and Grundkiewicz [87] adopt this multi-source model for automatic post-editing of MT output, and describe the computation of the decoder start state s_0 for the dual-source model as the concatenation of the averaged encoder contexts. The decoder consists of doubly-attentive cGRU cells, the only difference from the original conditional GRU being once again the concatenation of the context vectors.
- Tying of embedding matrices is possible. Press and Wolf [88] define tied embeddings as the weight tying of input and output embeddings of the decoder,

describing its effect in NMT as the reduction of number of parameters of the model by less than half, without compromising the performance. Three-way weight tying is also allowed, where input and output embeddings of the decoder, and the input embedding of the encoder are tied.

The deep encoder-decoder architectures implemented in Marian are explained in the following subsections.

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. [2].

The deep transition encoder is a bidirectional RNN, where the encoder recurrence depth is represented with L_s. The forward state of the i-th source word $\vec{h}_i \equiv \vec{h}_{i,L_s}$ is calculated such that, input of the first GRU transition is the word embedding x_i, whereas the other GRU transitions have no external inputs. The previous word state $\vec{h}_{i-1,L_s}$ is input to the first GRU transition for the current source word, enabling recurrence.

$$\begin{aligned} \vec{h}_{i,1} &= GRU_1\,(x_i, \vec{h}_{i-1,L_s}) \\ \vec{h}_{i,k} &= GRU_k\,(0, \vec{h}_{i,k-1}), \text{where } 1 < k \leq L_s \end{aligned} \tag{4.7}$$

Reverse source word states are calculated in a similar fashion, and concatenated to the forward source word states, forming the bidirectional source word states $C \equiv$

$\{[\overrightarrow{h}_{i,L_s}\ \overleftarrow{h}_{i,L_s}]\}$.

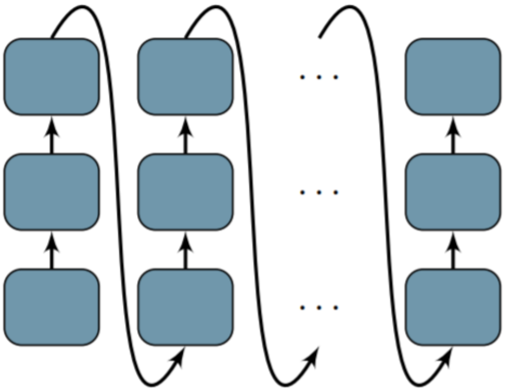


Figure 4.3. Deep transition decoder [2] (License provided in Appendix A)

The deep transition decoder (Figure 4.3) is an extension of the baseline decoder that consists of a transition depth of two, where GRU_1 takes in the embedding of the previous target word, and GRU_2 receives a context vector computed by the attention mechanism. This scheme is extended, so that the transition depth (decoder recurrence depth) becomes an arbitrary L_t, where the embedding of the previous target word is denoted as y_{j-1}, and the context vector computed by the attention mechanism as $ATT(C, s_{j,1})$:

$$\begin{aligned}
s_{j,1} &= GRU_1\left(y_{j-1}, s_{j-1,L_t}\right) \\
s_{j,2} &= GRU_2\left(ATT(C, s_{j,1}), s_{j,1}\right) \\
s_{j,k} &= GRU_k\left(0, s_{j,k-1}\right) \text{ where } 2 < k \leq L_t
\end{aligned} \tag{4.8}$$

Only the first two GRU transitions have external inputs. Finally, prediction of the current target word is achieved by the feed-forward output network, exploiting the target word state vector $s_j \equiv s_{j,L_t}$.

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 $L_t = 8$ has been adopted in all of the attentional encoder-decoder experiments except the final models (Section 5.4). Tied embeddings (weight tying of all embeddings and output layer) [88] have been employed to reduce the number of parameters. To reduce training time, layer normalization [89], an alternative to batch normalization has been 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 training cases (Figure 4.4). 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 has been applied along the RNN layers.

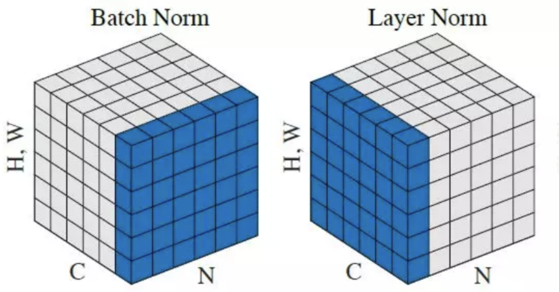


Figure 4.4. Batch normalization versus layer normalization [3]. Feature map tensors are shown, where pixels in blue are normalized. N: batch axis, C: channel axis, (H, W): spatial axes. (License provided in Appendix B)

Taking example from UEDIN's WMT18 system [69], Adam [90] has been used for the optimization of the models, with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. Learning rate was started at 0.0003 during training. Exponential smoothing, gradient clipping and for regularization, label smoothing [91] (0.1) as a way of encouraging the model to be less confident have been incorporated. The models have been trained on 2 GPUs on TÜBİTAK ULAKBİM's computing infrastructure, TRUBA (Turkish National e-Science e-Infrastructure), with mini-batch size fit into 9.5GB of GPU memory. Early stopping with a patience of 5 has been selected as the stopping criterium, 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 convergence. Best models according to 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 is explained 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, so that at each time step the bottom GRU takes external inputs from the outside, while the higher GRU is fed as external input, the state output of the one below it. 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 [2].

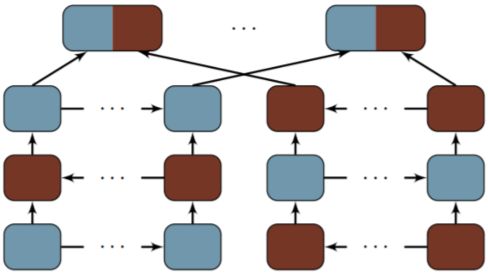


Figure 4.5. Alternating stacked encoder [2] (License provided in Appendix A)

A variation of Zhou et al.'s [66] LSTM-based model, the encoder is called an alternating stacked encoder (Figure 4.5). The forward encoder consists of a stack of GRUs, operating in alternating directions, the first GRU processing words in the forward direction, the second GRU in the backward direction, and so on. Assuming an encoder stack depth as D_s and the source sentence length N, the forward source word state $\vec{w}_i \equiv \vec{w}_{i,D_s}$ is obtained as:

$$
\begin{aligned}
\overrightarrow{w}_{i,1} &= \overrightarrow{h}_{i,1} = GRU_1\left(x_i, \overrightarrow{h}_{i-1,1}\right)\\
\overrightarrow{h}_{i,2k} &= GRU_{2k}\left(\overrightarrow{w}_{i,2k-1}, \overrightarrow{h}_{i+1,2k}\right)\\
&\quad \text{for } 1 < 2k \leq D_s\\
\overrightarrow{h}_{i,2k+1} &= GRU_{2k+1}\left(\overrightarrow{w}_{i,2k}, \overrightarrow{h}_{i-1,2k+1}\right)\\
&\quad \text{for } 1 < 2k+1 \leq D_s\\
\overrightarrow{w}_{i,j} &= \overrightarrow{h}_{i,j} + \overrightarrow{w}_{i,j-1}\\
&\quad \text{for } 1 < j \leq D_s
\end{aligned}
\tag{4.9}
$$

$\overrightarrow{h}_{0,k}$ and $\overrightarrow{h}_{N+1,k}$ are assumed to be zero vectors. At each level above the first, the word state of the current level $\overrightarrow{w}_{i,j}$ is computed as the sum of the word state of the previous level $\overrightarrow{w}_{i,j-1}$ and the current GRU cell's recurrent state $\overrightarrow{h}_{i,j}$, indicating residual connections.

The backward encoder operates the same way, where the words are processed backwards in the first level, and the rest of the levels alternate directions. The concatenation of the forward and backward word states constitutes the bidirectional word states $C \equiv [\overrightarrow{w}_{i,D_s}\ \overleftarrow{w}_{i,D_s}]$.

The stacked decoder also consists of stacked GRUs that do not alternate directions, but operate in the forward direction. The base GRU is a conditional GRU (cGRU) with a transition depth of two, whereas the higher RNNs are simple GRUs with residual connections and a transition depth of one. The target word states for decoder stack depth D_t are computed as follows for the higher GRUs:

$$\begin{aligned}
s_{j,1,1} &= GRU_{1,1}\left(y_{j-1}, s_{j-1,1,2}\right) \\
c_{j,1} &= ATT(C, s_{j,1,1}) \\
s_{j,1,2} &= GRU_{1,2}\left(c_{j,1}, s_{j,1,1}\right) \\
r_{j,1} &= s_{j,1,2} \\
s_{j,k,1} &= GRU_{k}\left(r_{j,k-1}, s_{j-1,k,1}\right) \\
r_{j,k} &= s_{j,k,1} + r_{j,k-1} \\
&\text{for } 1 < k \leq D_t
\end{aligned} \tag{4.10}$$

4.1.3. BiDeep Architecture

The BiDeep RNN is a novel architecture proposed by Miceli Barone et al. [2] as a mixture of deep transition and stacked architectures. D_s individually recurrent GRUs of the stacked encoders and decoders are replaced with multi-layer deep transition cells consisting of L_s GRU transition blocks. Hence, for the BiDeep RNN, the GRU_k in Equations 4.9, 4.10 is replaced with a multi-layer deep transition GRU: $DTGRU_k$, other computations remaining the same. The multi-layer $DTGRU_k$ cell is computed as:

$$\begin{aligned}
v_{k,1} &= GRU_{k,1}\left(in_k, state_k\right) \\
v_{k,t} &= GRU_{k,t}\left(0, v_{k_t-1}\right) \text{ for } 1 < k \leq L_s \\
DTGRU_k\left(in_k, state_k\right) &= v_{k,L_s}
\end{aligned} \tag{4.11}$$

In this research, the final models carry the BiDeep RNN architecture implemented with Marian, with 4 encoder layers (each with 2 transitional GRU cells) and 4 decoder layers (the first layer with 4 and the next layers with 2 transitional GRU cells). 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 as the deep transition models described in Section 4.1.1, with the same stopping criterion, trained on 2 GPUs.

4.2. Transformer Model

The sequential nature of the recurrent encoder-decoder models with attention makes parallelization within training examples difficult, especially for longer sentences. In addition, distant items may not affect each other's output without passing through many RNN steps or convolutional layers. In order to address these problems, Vaswani et al. [4] 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 a Transformer, a model that allows more parallelization, better translation quality and less training time.

The Transformer (Figure 4.6) uses stacked self-attention and point-wise, fully connected layers for the encoder and the decoder, where the encoder consists of 6 stacked layers, each with a multi-head self-attention layer and a position-wise fully connected feed-forward layer. Residual connection and layer normalization are applied around each two sub-layers. The decoder also consists of 6 stacked layers. In addition to the two sub-layers in the encoder, the decoder applies multi-head attention to the output of the encoder. Once again, residual connection and layer normalization are applied. The multi-head attention layer in the decoder is masked, to ensure that attention does not focus on unknown outputs in subsequent positions.

Vaswani et al. [4] define attention as a function of a query and a set of key-value pairs. A weight corresponding to the value is computed with a compatibility function of the key and the query. Two self-attention mechanisms have been introduced: Scaled Dot-Product Attention and Multi-Head Attention, explained in detail in Sections 4.2.1

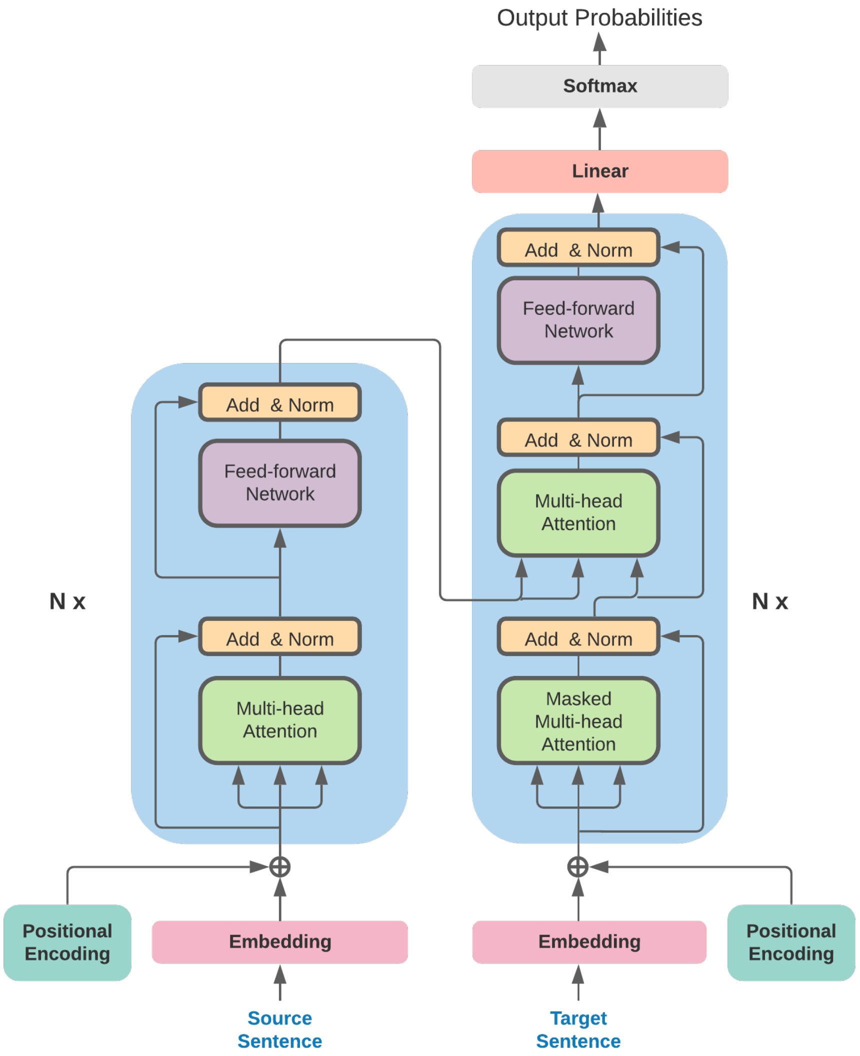


Figure 4.6. Transformer model architecture [4] (self-drawn)

and 4.2.2.

4.2.1. Scaled Dot-Product Attention

In Scaled Dot-Product Attention (Figure 4.7), queries and keys are of dimension d_k, and values of dimension d_v. The weights of the values are computed by taking the dot product of the query with all keys, divided by $\sqrt{d_k}$, and then applying softmax. The matrix of outputs is (queries, keys and values are packed into the matrices Q, K, V):

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

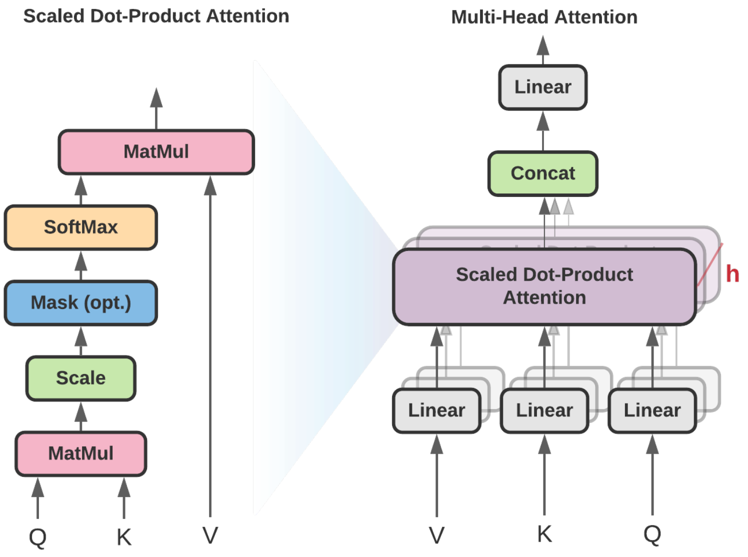


Figure 4.7. Self-attention models [4] (self-drawn)

The scaling of the dot product by $(1/\sqrt{d_k})$ is to prevent the dot product from growing too large.

4.2.2. Multi-Head Attention

In Multi-Head Attention, the queries, keys and values are linearly converted h times with different, learned projections into d_k, d_k and d_v dimensions. Afterwards, the attention function is performed in parallel, generating d_v-dimensional output values. These output values are then concatenated and projected, yielding the final values (Figure 4.7).

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

The advantage of the multi-head attention is jointly obtaining information from different representation subspaces at different positions. Vaswani et al. [4] exploited the multi-head attention in three ways. Firstly, as in the encoder-decoder attention models, as a way for the decoder to focus on specific parts of the input, secondly, inside the encoder, and thirdly, inside the decoder.

The Marian implementation of the Transformer, following the explained approach, was used in this study for the Transformer models. Encoder and decoder depths have both been set to 6 layers, employing 8-head multi-head attention. All Transformer models have been trained on 4 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, size of the position-wise feed-forward network has been set to 4096 instead of 2048, and the size of embedding vector has been set to 1024 instead of 512, resembling Google's Transformer-Big architecture. Although compromising from speed and memory usage, improvement over the original has been observed (Section 5.1).

In addition to dropout between Transformer layers (0.1), dropout 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. In order to be compatible with the increase in parameters, mini-batch size was fit into 8GB of GPU memory. Best models according to 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 usage of monolingual corpora, through self-training for source-side, and through copying and back-translation for target-side parallel data augmentation. The Turkish and English monolingual data has been obtained from the WMT News Crawl 2020 dataset.

For experimentation on different input variations, a shallow Turkish-English attentional encoder-decoder model has been trained using only the 207K SETimes-clean parallel corpus. The Moses scripts for tokenization, truecasing and punctuation normalization [83] have been applied to the parallel corpus. Joint byte pair encoding (BPE) has been employed for subword segmentation [70]. With the trained model, source-side (Turkish) monolingual data of 450K sentences has been translated into English. The combination of the parallel SETimes corpus and the synthetic corpus have been cleaned with the Moses script, resulting in Corpus A with 656K sentences (Table 4.1).

Table 4.1. Statistics of augmented corpora after tokenization and cleaning.

	Synthetic				
Corpus	**Self-trained**	**Back-translated**	**Copied**	**Original**	**Total**
SETimes-clean	-	-	-	207,373	207K
A	448,811	-	-	207,373	656K
B	1,994,892	-	-	207,373	2.2M
C	2,483,765	-	-	207,373 x 5	3.5M
D	-	2,404,835	-	207,373 x 5	3.4M
E	-	2,404,835	981,141	207,373 x 5	4.4M
F	2,483,765	2,404,835	981,141	207,373 x 5	6.9M

In order to observe how the amount of synthetic data affects translation quality, source-side (Turkish) monolingual data of 2M sentences has been translated into English in the same manner. Mixed with the SETimes-clean parallel corpus, Corpus B (2.2M sentences) has been used for training attentional encoder-decoder and Transformer models, with different input segmentation techniques.

After the results of different input variations have been obtained, the best input segmentation method has been selected for the final models. The final models were trained on a combination of synthetic self-trained data (by translating source-side monolingual data), copied data, and synthetic back-translated data (by translating target-side monolingual data).

For the final models, 2.5M Turkish sentences from the WMT News Crawl 2020 dataset have been translated by a Turkish-English attentional encoder-decoder model with BiDeep architecture, trained on only the SETimes-clean corpus. Pre-processing of the parallel corpus differed from the aforementioned process, where the best input segmentation method, Morphemes has been applied, which will further be explained in Section 4.4. The 2.5M synthetic parallel corpus has undergone special cleaning steps, taking example from Durgar El-Kahlout et al. [92]. A sentence pair has been removed if:

- the synthetic sentence consists of only one word
- token count in synthetic sentence / token count in authentic sentence > 3
- a token in the synthetic sentence repeats itself 3 times consecutively

After cleaning is complete, the synthetic corpus has been paired with the SETimes-clean corpus. In order to prevent the ratio of synthetic over original from becoming too much in favor of the synthetic, the original parallel corpus has been oversampled (copied) 5 times (shown in Table 4.1 as x 5), forming Corpus C (3.5M sentences).

In addition to exploiting source-side monolingual data via self-training, target-side monolingual data has also been incorporated via back-translation. An English-Turkish BiDeep model has been trained on only the SETimes-clean corpus, undergoing the same pre-processing operation as the source-side. Afterwards, 2.5M English sentences from the WMT News Crawl 2020 dataset have been back-translated. The obtained synthetic back-translated corpus has been cleaned with respect to the aforementioned three conditions. Corpus D consists of the clean back-translated corpus, and SETimes-clean, oversampled 5 times (3.4M sentences).

Following the work of Currey et al. [46], a copied corpus of 1M sentences has been created. 1M English sentences from the WMT News Crawl 2020 dataset has been taken, and a bitext has been formed, with the source-side identical to the target-side. The copied corpus has been added to Corpus D, to form Corpus E (4.4M sentences). In addition, a corpus that contains all the augmentations (the copied, the back-translated and the source-side translated corpora) has been put together, to form Corpus F (6.9M sentences).

4.4. Input variations

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

Input segmentation methods are explained with examples and statistics in the following subsections. 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. Byte Pair Encoding

Byte pair encoding (BPE) is a word segmentation algorithm that encodes rare words via subword units [70]. The open vocabulary problem is tackled by creating a fixed-size vocabulary consisting of variable-length character sequences. Additionally, 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 research, joint BPE is applied, which is learning the encoding on the union of source and target vocabularies, observed to improve consistency between source and target segmentation. Joint BPE learning is achieved by the concatenation of source and target corpora, and then applying the *subword-nmt* tool [71] on the concatenated corpora. Number of merge operations has been set to 85,000. Segmented subwords carry the "@@" symbol at the end, except for the rightmost subword of a word (see example in Table 4.2).

Table 4.2. Examples of Byte Pair Encoding (BPE) and WordPiece (BERT) segmentation.

	EN	**TR**
Original Sentence	Unfortunately, Greece as a full member of the Alliance, threatens to use its veto.	Kendi zayıflığımız yüzünden bu hedeflere geçtiğimiz yıl ulaşamamış olmaktan üzüntü duyuyorum.
BPE Segmented Sentence	Un@@ fortun@@ ate@@ ly , Greece as a full member of the Alliance , threatens to use its veto .	kendi zayıf@@ lığımız yüzünden bu hedeflere geçtiğimiz yıl ulaş@@ amamış olmaktan üzüntü duyuyorum .
BERT Segmented Sentence	Un ##fort ##unate ##ly , Greece as a full member of the Alliance , threatens to use its veto .	kendi zayıf ##lığımız yüzünden bu hedeflere geçtiğimiz yıl ulaşama ##mış olmaktan üzüntü duyuyorum .

4.4.2. WordPiece

WordPiece algorithm is a word segmentation algorithm similar to BPE [7]. 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 [93] implementation of BERT's [55] WordPiece tokenizers have been used. BERT Transformer models are pre-trained on large English and Turkish data, with masked language modeling and next sentence prediction objectives, thus learning an inner representation on the languages, afterwards to be used for fine-tuning. BERT tokenizers are created with the WordPiece algorithm, and can be used for tokenizing data for fine-tuning BERT language models, or any given task.

For English, the case-sensitive *bert-base-cased* tokenizer [55] with a vocabulary size of 28,996, and for Turkish the *distilbert-base-turkish-cased* tokenizer [94] with a vocabulary size of 32,000 has been used, which is a distilled, lighter version of BERT [95]. After separately segmenting the Turkish and English sentences, subwords carry the "##" symbol at the beginning, except for the leftmost subword of a word (see example in Table 4.2).

Although being quite similar to the BPE algorithm, BERT's tokenizer benefit from being pre-trained 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.

4.4.3. Morphemes and Allomorphs

Morphemes are defined as small lexical items that carry a meaning. Free morphemes can function alone with a specific meaning, whereas bound morphemes function as parts of words, used in conjunction with a root or other bound morphemes. Allomorphs are different phonological variants of morphemes. The difference can be in spelling or pronunciation. For example, the ablative morpheme in Turkish is *"DAn"*, which has four allomorphs, depending on the root word it is attached to: *"tan, ten, dan, den"*.

In this research, complex morphology of the Turkish language has led to the idea of morphologically motivated input segmentation, meaning the breaking up of a word into its morphemes via morphological analysis and disambiguation. The morphosyntactic and morphosemantic information carried by the morphemes and allomorphs is expected to be leveraged with several approaches, and comparison within these methods and with mainstream word segmentation methods (BPE, WordPiece) that are not linguistically motivated has been carried out.

Before morphological analysis and disambiguation, Turkish sentences have been cleaned and truecased with the Moses scripts. Afterwards, the Zemberek tokenizer [6] has been used for the separation of words and punctuation.

4.4.3.1. Morphemes. Morphological analysis of Turkish sentences has been performed using Sak et al.'s [5] tool. After morphologically parsing the sentence, disambiguation is applied on all possible parses of a word, selecting the best morphological analysis (Table 4.3).

After disambiguation, the sentence is reconstructed word by word, using a Python script. The disambiguated morphological analysis of each word is used to extract its morphemes, separated by a space. Each morpheme after the root morpheme starts with an underscore ("_"). Special extraction is used for capital letters from the original corpus, since this knowledge is lost during analysis.

The second input variation with this approach is obtained by concatenating the morphemes other than the root, called Concatenated Morphemes. In this case, the concatenated morpheme sequence carries an underscore at the beginning. The third input variation is using only the root and the morpheme at the end of the word, referred to as Last Morpheme. The reason for the usage of the last morpheme is based on Oflazer et al.'s [96] observation that syntactic relation links are usually associated with the last morpheme/IG (inflectional group) of a word. This final input segmentation method results in syntactic and semantic loss, but decreases the amount of morphemes in a sentence, a rather interesting method to observe. All segmentations with morphemes are exemplified in Table 4.5.

Table 4.3. Morphological analysis and disambiguation (Sak et al. [5]) 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)

4.4.3.2. Allomorphs. Usage of morphemes in the input representation has the effect of vocabulary reduction, since different phonetic variations of suffixes, affixes and roots are represented with a single form. With the intention of observing if the drop in vocabulary size improves translation quality, allomorphs, that is to say morphemes with phonological variations, have also been investigated. However, the morphological analysis and disambiguation tool of Sak et al. does not provide this functionality. To achieve an allomorph segmentation, the Zemberek tool [6] has been used.

Table 4.4. Morphological analysis and disambiguation (Zemberek [6]) 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)

Morphological analysis and disambiguation with Zemberek operates in a similar way as Sak et al., where allomorphs instead of morphemes are output (Table 4.4). Reconstruction of the sentence with subword units is the same as described in Section 4.4.3.1. Allomorph segmentation is extended with two additional methods: Concatenated Allomorphs and Last Allomorph, examples of which are presented in Table 4.5.

4.4.4. Morphological Tags

Usage of morphological tags instead of surface forms of the morphemes has been previously employed by researchers in the Turkish-English MT task [77, 79]. In order to extract morphological tags of a word, morphological analysis and disambiguation [5] has been performed as described in Section 4.4.3.1 for Morphemes. Two different segmentation methods have been employed.

4.4.4.1. Morphological Tags in Word. In this scenario, only the morphological tags that correspond to a lexical item inside the word have been included in the input representation. The root word has been represented in its surface form, afterwards to be continued by the morphological tags of the rest of the morphemes. The Morph-Tags in Word setting was expected to produce a similar translation performance as Morphemes, with minor differences, due to the fact that morphological tags have almost one-to-one correspondence with morphemes: YH ↔ [Acc], lAr ↔ [A3pl], NHn ↔ [Gen].

4.4.4.2. All Morphological Tags. All morphological tags have been included in the input representation, including the type of root tag (Noun, Verb, Adj, etc.). This segmentation method (All Morph-Tags) significantly increases the number of tokens in a sentence, and is thus expected to yield deteriorated results.

Examples of input segmentation using morphological tags are presented in Table 4.5.

4.4.5. Multi-source

As mentioned, 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.

Table 4.5. Examples of input variations of the sentence:

“Gün geçtikçe bu tarz haberleri daha sık duyar hale geldik .”

Input Variation	**Segmented Sentence**
Morphemes	Gün geç _DHkçA bu tarz haber _lAr _SH daha sık duy _Ar hâl _YA gel _DH _k .
Concatenated Morphemes	Gün geç _DHkçA bu tarz haber _lArSH daha sık duy _Ar hâl _YA gel _DHk .
Last Morpheme	Gün geç _DHkçA bu tarz haber _SH daha sık duy _Ar hâl _YA gel _k .
Allomorphs	Gün geç _tikçe bu tarz haber _ler _i daha sık duy _ar hal _e gel _di _k .
Concatenated Allomorphs	Gün geç _tikçe bu tarz haber _leri daha sık duy _ar hal _e gel _dik .
Last Allomorph	Gün geç _tikçe bu tarz haber _i daha sık duy _ar hal _e gel _k .
Morph-Tags in Word	Gün geç Adv_AsLongAs bu tarz haber A3pl P3sg daha sık duy Aor hâl Dat gel Past A1pl .
Morph-tags-all	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 hâl Noun NoHats A3sg Pnon Dat gel Verb Pos Past A1pl . Punc

Pan et al. have trained a multi-source NMT model with a word-based encoder to capture word features, and a knowledge-based encoder to capture linguistic features [80]. Similar to this approach, two input variations, Allomorphs and Morph-Tags in Word have been used together to train a single multi-source attentional encoder-decoder model, the former entailing morphemes in their surface form, the latter carrying their morphological tags, thus clarifying the syntactic and semantic purpose of a morpheme inside the sentence.

After applying morphologically motivated segmentation methods (Morphemes, Allomorphs, Morphological Tags and Multi-source), Turkish words are broken into smaller lexical items. However, rare words, or proper nouns that cannot be recognized by the morphological analyzer are left unsegmented. Thus, for each linguistically motivated input variations, the segmented input representation is further segmented via BPE, improving translation quality for all cases (Table 5.3).

In order to numerically distinguish the effect of each input segmentation method on the Turkish corpus, token counts, unique token counts and average sentence lengths have been provided in Table 4.6. The given statistics are for Corpus A, which consists of 207K parallel and 449K synthetic sentences (total 656K sentences). Morphologically motivated input variations are given solely and with further BPE segmentation (indicated as +BPE), to observe the drop in vocabulary size (unique tokens) and increase in average sentence length. Minor numerical differences between and within morphological variations are due to the differences of morphological analyzers and/or due to some minor exceptions missed by the Python scripts that create each corpus.

Some observations that can aid while analyzing the experimental results are:

- 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 BERT, due to the fixed vocabulary size of the Turkish BERT tokenizer (32,000).
- Concatenation of morphemes/allomorphs and taking the last morpheme/allomorph decreases the average sentence length, compared to using all morphemes/allomorphs separately.
- Applying BPE segmentation after morphologically motivated input segmentation reduces the vocabulary size, by separating proper nouns, rare or long words that could not be segmented by the morphological analyzer.
- Using all morphological tags of a word (Morph-tags-all) more than doubles the average sentence length, thus lowering the expectation for high translation quality for this variation.

Table 4.6. Statistics of input variations on Corpus A. +BPE indicates further BPE segmentation after the morphologically motivated input variation is applied.

				+ BPE		
Input Variation	**Tokens**	**Unique Tokens**	**Average Sentence Length**	**Tokens**	**Unique Tokens**	**Average Sentence Length**
Unsegmented	12,409,844	396,654	18.91	-	-	-
BPE	14,376,964	66,713	21.91	-	-	-
BERT	15,579,083	29,225	23.74	-	-	-
Morphemes	20,567,955	154,087	31.34	20,810,209	63,751	31.71
Concatenated Morphemes	17,121,720	162,108	26.09	17,388,020	65,028	26.50
Last Morpheme	17,159,258	154,110	26.15	17,401,577	63,739	26.52
Allomorphs	20,559,065	160,029	31.33	20,825,083	64,177	31.74
Concatenated Allomorphs	17,203,831	169,077	26.22	17,489,665	66,012	26.65
Last Allomorph	17,173,915	152,364	26.17	17,410,779	63,718	26.53
Morph-Tags in Word	20,570,495	148,191	31.35	20,791,690	63,163	31.69
Morph-tags-all	49,245,643	148,217	75.05	49,466,709	63,125	75.39

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.

4.5. Ensemble and Rescoring

In this extensive study on Turkish-English NMT, systems with various model architectures, data augmentation methods and input variations have been trained, and comparison between different settings has been manifested. However, 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 [97], model ensembling has been utilized during all experimentations. Moreover, the observation of Liu et al. [22] of the imbalance in output sentences (better translation quality of prefixes over suffixes) has encouraged the use of bidirectional decoding via rescoring [23].

Model ensembling in this research is carried out by training multiple models with different random initializations of model parameters. For all experiments except the final models, 4 left-to-right (L2R) and 4 right-to-left (R2L) models have been trained, each of the four models randomly initialized with different seeds. For the final models, the number of models for each direction has been decreased to 2, due to time and memory concerns on account of the largeness of training corpora.

The same training corpus is used to train the L2R and R2L models. After training is complete, decoding is performed in such a way that, 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, each 50 hypotheses of each test sentence is rescored with the R2L models (input sentence and the hypothesis are fed to the R2L models for scoring). The hypothesis that obtains the highest score from the sum of L2R and R2L model scores is selected as the final translation.

5. EXPERIMENTS AND RESULTS

Neural machine translation 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 tremendously 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, the best model architectures, data augmentation and input segmentation techniques are selected to train the final models.

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.

5.1. Model Architectures

Among neural architectures in NMT, attentional encoder-decoder and Transformer architectures are the most widely adopted, 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 research, both attentional encoder-decoder and Transformer models have been experimented with, with different input representations and hyperparameters, so as to deduct the most suitable architecture for the low-resourced Turkish-English language pair.

Encoder-decoder models have been constructed with deep transition architecture, with an embedding size of 512, and an RNN hidden state size of 1024. The encoder and the decoder have a single layer, with an encoder recurrence depth of $L_s = 4$ and a decoder recurrence depth of $L_t = 8$.

Transformer models carry 6 layers for the encoder and the decoder. Size of the position-wise feed-forward network of the Transformer model was set to 2048, and the embedding size to 512. However, this architecture proved not to be adequate for the low-resource scenario of this task, yielding a 2-3 drop in BLEU score with respect to the deep transition model. Therefore, from this point on the Transformer models have been trained with a feedforward network size of 4096, and an embedding size of 1024, resembling Google's Transformer-Big architecture.

For model architecture experiments, all models have been trained with Corpus A, consisting of 207K original, and 449K synthetic parallel sentences (a total of 656K). Separate systems have been trained with BPE and BERT (WordPiece) input representations. For each different architecture, 4 left-to-right (L2R) and 4 right-to-left (R2L) models have been trained. Translation on WMT17 and WMT18 test sets have been done separately for each L2R and R2L model, and the average BLEU scores of 4 L2R, and 4 R2L models have been reported. The final system for each architecture, is an ensemble of 4 L2R models, which produce 50-best translation hypotheses (beam size 50), which are in turn rescored by 4 R2L models. BLEU-cased scores of each system have been provided in Table 5.1.

The shallow Turkish-English NMT model used for the automatic translation of the source monolingual data was trained solely on the 207K SETimes corpus, with BPE segmentation. This single model has yielded 15.12 BLEU for *newstest2017*, and 15.64 BLEU for *newstest2018*. After data augmentation with synthetic 449K sentences, the systems trained on Corpus A outperform the baseline shallow NMT model trained on the SETimes corpus by 1-3 BLEU, except for Transformer-BPE with network size 512, 2048, which shows very poor translation performance.

Table 5.1. TR-EN news translation (BLEU-cased) scores of systems with different model architectures.

				newstest2017			newstest2018		
Model	**Input**	**No. layers**	**Network Size**	**L2R Avg.**	**R2L Avg.**	**Ensemble**	**L2R Avg.**	**R2L Avg.**	**Ensemble**
Deep Transition	BPE	1(4), 1(8)	512, 1024	16.09	16.68	17.46	16.72	17.31	18.23
Deep Transition	BERT	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	BERT	6, 6	1024, 4096	**16.93**	**17.19**	**18.14**	**17.47**	**17.62**	**18.70**

The positive effect of model ensembling can be observed for each system, increasing the BLEU score by up to 1.5. The Transformer architecture with network size (1024, 4096), yields better results than the deep transition encoder-decoder architecture for both input representations. An interesting deduction is that, BERT input representation over BPE improves the L2R average of systems, yet deteriorates or very slightly improves translation quality of the R2L average. Furthermore, the Transformer architecture seems to be reacting better to BERT with respect to BPE, than the encoder-decoder.

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

5.2. Data Augmentation

A comprehensive 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. Different data augmentation techniques have been taken into consideration, and synthetic parallel data has been obtained through self-training with source-side monolingual data. The effect of data augmentation through self-training has been observed with different model architectures and input representations, for the purpose of enlightening how the low-resource setting can be eliminated.

Transformer (with network sizes 1024 and 4096) and deep transition encoder-decoder models have been trained as described in the model architecture experiments. Each model architecture has been experimented with BPE and BERT input representations. Two training corpora have been 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 of 4 L2R and 4 R2L models, and also of the final ensemble models have been presented in Table 5.2.

Table 5.2. 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	BERT	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	BERT	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

It is important to note that the source-side (Turkish) 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-BERT and Transformer-BERT systems 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 synhtetic data is increased to 4.5 times its size, the Transformer-BPE system receives 0.48 and 0.8 higher BLEU scores for the WMT17 and WMT18 test sets, respectively.

5.3. Input Variations

The rich morphology of Turkish, and the scarceness of data has led to the investigation of morphologically motivated input segmentation methods with respect to more general input representations, like BPE and BERT (WordPiece). Linguistically motivated input representations proposed and compared in this research are: Morphemes and Allomorphs, each used separately, concatenated or by taking the last subword after the root; Morphological Tags, by using only the tags that correspond to a lexical item within the word, or using all tags; and Multi-source, by using both Allomorphs and Morphological Tags in Word to train a single encoder-decoder model.

After applying morphologically motivated input segmentation to the tokenized Turkish corpus, BPE algorithm is applied, further to segment rare words or proper nouns that could not be recognized by the morphological analyzer. The positive effect of BPE segmentation on top of morphological decomposition is demonstrated in Table 5.3. For all input variations, a left-to-right (L2R) model was trained on an input without further BPE segmentation. The translation result of this model is compared to an average of 4 L2R models trained with further BPE segmentation. The BLEU scores showed 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, apart from this experiment, all mentioned morphologically motivated input decomposition methods are supported with further BPE segmentation.

Table 5.3. Left-to-right 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.**
Morphemes	16.29	↑ 17.21
Concatenated Morphemes	16.28	↑ 17.28
Last Morpheme	15.08	↑ 16.03
Allomorphs	16.19	↑ 17.19
Concatenated Allomorphs	15.87	↑ 17.06
Last Allomorph	14.98	↑ 16.11
Morph-Tags in Word	16.49	↑ 17.25
Morph-tags-all	14.91	↑ 15.49
Multi-source	16.43	↑ 17.32

All input variations have been used to train a deep transition attentional encoder-decoder model, with network sizes 512 (embedding size) and 1024 (RNN hidden state size). As described for previous experiments, a single-layer encoder with recurrence depth $L_s = 4$ and a single-layer decoder with recurrence depth $L_t = 8$ have been employed. The systems were trained on Corpus A (656K sentences: 207K original, 449K synthetic), to keep the systems safe from changes in training corpus quantity and quality. For all input segmentations, 4 L2R and 4 R2L models have been trained, and their average BLEU scores are reported. The final system is an ensemble of 4 L2R models that produces an n-best list of 50 translation hypotheses, which is rescored by the 4 R2L models. BLEU-cased scores of each system are shown in Table 5.4.

Table 5.4. TR-EN news translation (BLEU-cased) scores of systems with different input segmentation methods.

	newstest2017			**newstest2018**		
Input	**L2R Avg.**	**R2L Avg.**	**Ensemble**	**L2R Avg.**	**R2L Avg.**	**Ensemble**
BPE	16.09	16.68	17.46	16.72	17.31	18.23
BERT	16.23	16.46	17.63	16.82	17.08	18.26
Morphemes	**16.71**	**17.35**	**18.42**	17.21	**17.78**	**18.83**
Concatenated Morphemes	16.65	17.17	18.16	17.28	17.68	18.79
Last Morpheme	15.62	16.03	17.14	16.03	16.39	17.39
Allomorphs	16.64	17.13	18.08	17.19	17.41	18.66
Concatenated Allomorphs	16.48	16.89	18.11	17.06	17.58	18.65
Last Allomorph	15.78	16.17	17.44	16.11	16.53	17.58
Morph-Tags in Word	16.52	17.09	18.24	17.25	17.71	18.58
Morph-tags-all	14.91	15.01	16.29	15.49	15.60	17.02
Multi-source	16.38	16.90	18.21	**17.32**	17.57	18.57

Before analyzing the input segmentation results, a reference Turkish word ("evdekilerle", translated into English as "with the ones at home") is selected, in order to aid in understanding how each input representation looks like.

Among non-linguistically motivated methods, BERT (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 BERT and BPE, which is promising for the low-resource Turkish-English language pair.

The best input segmentation method is selected to be Morphemes (ev _DA _ki _lAr _YlA) improving the BLEU score by 0.96 with respect to BPE for *newstest2017*, and 0.5 for *newstest2018* in the final ensemble results.

Among the Morphemes approach, the best method is using all morphemes separately (ev _DA _ki _lAr _YlA) instead of concatenating (Concatenated Morphemes: ev _DAkilArYlA) or taking the last morpheme after the root (Last Morpheme: ev _YlA).

Allomorphs (ev _de _ki _ler _le) method yields a BLEU score in between BPE and Morphemes. This performance drop can be explained with the vocabulary reducing effect of Morphemes, due to the elimination of phonetic variations within roots, suffixes and affixes. In addition, usage of different morphological analyzers and disambiguators (Sak et al. vs. Zemberek) may also explain the difference, requiring further comparison on their performances.

Taking the last allomorph after the root (Last Allomorph: ev _le) seems to outperform the Last Morpheme approach, yet is far from competing with the non-linguistically motivated BPE or BERT, observed especially in the *newstest2018* final ensemble result. However, using only the last (rightmost) allomorph after the root 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 correlation between morphological tags and morphemes in Sak et al.'s morphological analyzer, the Morph-Tags in Word (ev Loc Adj_Rel A3pl Ins) approach does not achieve the same translation performance as Morphemes. Being an approach adopted by researchers in the Turkish-English NMT task, it is useful to realize that using morphemes instead of their morphological tags turns out to be more successful.

The morphological analyzer and disambiguator produce beneficial information, including type of nouns, meaning, purpose and person of the morphemes, etc. Incorporating all of this information was intended to be achieved, by including all morphological tags that correspond to a word (Morph-tags-all: ev Noun A3sg Pnon Loc Adj_Rel A3pl Pnon Ins). However, this approach resulted in unnecessarily long sentences, growing the average sentence length drastically. Thus, the BLEU score dropped from 18.24 to 16.29 for *newstest2017*, and from 18.58 to 17.02 for *newstest2018*, with respect to Morph-Tags in Word.

Morphemes in their surface forms (Allomorphs) may not present all the syntactic and semantic information hidden within. However, this information can be obtained from the corresponding morphological tag (Morph-Tags in Word). The idea behind the multi-source setting is to use Allomorphs with a word-based encoder, and Morph-Tags in Word with a knowledge-based encoder simultaneously, collecting as much information from a word as possible. However, the Multi-source input segmentation method (word represented with both "ev _de _ki _ler _le" and "ev Noun A3sg Pnon Loc Adj_Rel A3pl Pnon Ins") improved the final ensemble BLEU score of the Allomorphs for *newstest2017* only by 0.13, and failed to improve the score for *newstest2017*. Using one input representation is much more preferable with regard to memory and time consumption, thus proving not to be a good engineering choice for this task.

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 supports the argument of Liu et al. [22], regarding better translation of suffixes with the use of right-to-left decoding.

Determining the weaknesses and strengths of different input representations for the Turkish-English pair is meaningful, considering that scarce data makes it essential to represent the morphologically rich Turkish language as best as possible. After comprehensive analysis, Morphemes and Morph-Tags in Word on top of BPE are shown to be effective input segmentation methods, preferable to using only non-linguistically motivated methods, like BPE and BERT.

5.4. Final Models

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 high translation quality and generalization.

With regard to model architecture, it has been observed that the Transformer architecture outperforms the attentional encoder-decoder model. This observation is expected to be confirmed in the final models. Therefore, attentional encoder-decoder with BiDeep architecture has been employed, with 4 encoder layers (cell depth: 2) and 4 decoder layers (base cell depth: 4, higher cell depth: 2), as described in Section 4.1.3, with network sizes (512, 1024). The Transformer models carry 6 encoder and decoder layers with network sizes (1024, 4096).

Three approaches have been used for data augmentation in the final models: self-training (translating source-side monolingual data), back-translation (translating target-side monolingual data) and copying (forming a bitext with source-side identical to target-side). Selected as the best input segmentation method, Morphemes followed by BPE segmentation has been used as input for training the Turkish-English and English-Turkish BiDeep NMT models on the 207K parallel corpus, to form self-trained data from source-side and back-translated data from target-side monolingual data.

BiDeep and Transformer models have been trained on Corpus C (207K original data oversampled 5 times + 2.5M synthetic self-trained data) and Corpus D (207K original data oversampled 5 times + 2.4M synthetic back-translated data). Afterwards, only Transformer models were trained on Corpus E (Corpus D + 1M copied data) and Corpus F (Corpus D + 1M copied data + 2.5M synthetic self-trained data), due to time and resource limitations, relying on the observation that the Transformer model outperforms the BiDeep models for Corpus C and Corpus D.

For each system, 2 L2R and 2 R2L models have been trained. The average of Turkish-English translation results of the L2R and R2L models are provided in Table 5.5, evaluated for the WMT17 and WMT18 test sets. The final models are an ensemble of 2 L2R models that output 50-best translation hypotheses, rescored by 2 R2L models. State-of-the-art results for the test sets submitted by University of Edinburgh to WMT18 [69] have been given in the table for comparison.

The three bottom rows of Table 5.5 represent hybrid systems, that is to say, an ensemble of multiple systems. For the first hybrid system, 2 L2R BiDeep models and 2 L2R Transformer models trained on Corpus D have been ensembled. A total of 4 L2R models of the two Transformer models have created 50-best hypotheses, which were in turn rescored by the R2L models of the two systems (4 in total). In a similar fashion, the second hybrid system was formed as an ensemble of 2 L2R BiDeep models and 2 L2R Transformer models trained on Corpus D; and 2 L2R Transformer models trained on Corpus E. Rescoring is carried out by a total of 6 R2L models of these systems. For the third hybrid system, a total of 6 L2R Transformer models trained on corpora D, E and F have been used in ensemble to create the translation hypotheses. Rescoring has been done by the 6 R2L models of these systems.

The BiDeep Turkish-English NMT model used for the automatic translation of the source monolingual data was trained solely on the 207K SETimes corpus, with Morphemes segmentation. This single model has yielded 16.45 BLEU for *newstest2017*, and 16.77 BLEU for *newstest2018*. The benefit of data augmentation with synthetic self-trained 2.5M sentences is shown in the translation scores of the systems trained on Corpus C, which outperform the baseline NMT model trained on the SETimes corpus by 1.5-3 BLEU. The BiDeep English-Turkish NMT model that is utilized for back-translation with Morphemes segmentation, yielded an English-Turkish BLEU score of 22.57 for *newstest2017*, and 22.19 for *newstest2018*. The comparison of this baseline model to the final models would not be sensible, since the translation directions are opposite.

When the influence of self-training and back-translation on translation quality is evaluated, the power and effectiveness of back-translation can obviously be seen. With nearly the same amount of synthetic data, back-translation improves the BLEU score by 4.3 (*newstest2017*) and 5.25 (*newstest2018*) for the BiDeep model, with respect to self-training. Similarly, the Transformer model trained on Corpus D outperforms the Transformer model trained on Corpus C by 4.66 (*newstest2017*) and 6.53 (*newstest2018*) BLEU. 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 (Corpus D).

When all data augmentation methods are used together (Corpus F), the performance seems to fall to a BLEU score between the self-trained (Corpus C) and back-translated (Corpus D) 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, with a 26.21 BLEU on the WMT18 test set. The best translation performance for the WMT17 and WMT18 test sets is obtained from the hybrid of BiDeep and Transformer models trained on corpora D and E, with 24.74 and 26.38 BLEU, respectively. Even though the L2R and R2L averages of the hybrid systems are not necessarily higher than that 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 (Morphemes) shows its advantages in the given results, coming close to the state-of-the-art

Table 5.5. TR-EN news translation (BLEU-cased) scores of the final models.

		newstest2017			**newstest2018**		
Input	**Training Corpus**	**L2R Avg.**	**R2L Avg.**	**Ensemble**	**L2R Avg.**	**R2L Avg.**	**Ensemble**
UEDIN (2018) [69]	*3.5M back-translated + copied*	-	-	*26.6*	-	-	*28.2*
Bi-Deep	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
Bi-Deep	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 ensemble	D	22.04	22.00	24.37	23.47	23.30	26.21
BiDeep + Transformer ensemble	D, E	22.15	21.79	**24.74**	23.74	23.14	**26.38**
Transformer ensemble	D, E, F	21.67	21.35	24.58	23.14	22.40	25.86

by ≈1.8 BLEU. Further experimentation on different amounts of back-translated data, deeper Transformer architectures or more advanced ensembling methods are planned to be explored, and are left for future work.

6. CONCLUSION AND FUTURE WORK

This study approaches 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, though costly in memory and time consumption, outperforms the attentional encoder-decoder models.

Considering that the initial 207K Turkish-English parallel SETimes corpus is too small for training a deep NMT model with high translation performance, parallel data has been augmented through three methods. The first method is self-training, where a Turkish-English NMT model has been trained on the SETimes corpus, and source-side (Turkish) monolingual data has been translated into target-side (English). The second method is back-translation, where the process of self-training is in reverse, translating target-side (English) monolingual data into source-side (Turkish) with an English-Turkish NMT model. The third and final method is through copying target-side (English) data directly to the source-side, creating a bitext with identical source and target sides.

Initial 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. Since BPE input was fed into the NMT model that translated source monolingual data into target, models trained with BERT (WordPiece) input could not benefit from the increase in synthetic data created with a BPE-input model.

Coping with the low-resource setting of Turkish-English NMT has been aimed to be achieved, by extracting as much syntactic and semantic information from the input as possible. The rich morphology of Turkish has encouraged the usage of morphologically motivated input segmentation methods instead of more general approaches that can be applied to any language, and are statistically rather than linguistically motivated. With this motivation in mind, nine morphologically motivated input representation 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 Morphemes, outperforming BPE by 0.96 BLEU.

Final models have been trained with the BiDeep attentional encoder-decoder and Transformer architectures on data augmented corpora of up to 6.9M sentences, with input in the form of Morphemes + BPE. 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 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. 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. Contributions made to the Turkish-English NMT task are aimed to be extended to the English-Turkish direction, and also other language pairs containing Turkish.

REFERENCES

1. Bahdanau, D., K. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", Y. Bengio and Y. LeCun (Editors), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, `http://arxiv.org/abs/1409.0473`.

2. Miceli Barone, A. V., J. Helcl, R. Sennrich, B. Haddow and A. Birch, "Deep architectures for Neural Machine Translation", *Proceedings of the Second Conference on Machine Translation*, pp. 99–107, Association for Computational Linguistics, Copenhagen, Denmark, Sep. 2017, `https://aclanthology.org/W17-4710`.

3. Wu, Y. and K. He, "Group Normalization", *CoRR*, Vol. abs/1803.08494, 2018, `http://arxiv.org/abs/1803.08494`.

4. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is All you Need", I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (Editors), *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc., 2017.

5. Sak, H., T. Güngör and M. Saraçlar, "Morphological Disambiguation of Turkish Text with Perceptron Algorithm", A. Gelbukh (Editor), *Computational Linguistics and Intelligent Text Processing*, pp. 107–118, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

6. Ahmetaa, "ahmetaa/zemberek-nlp", `https://github.com/ahmetaa/zemberek-nlp`.

7. Wu, Y., M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, u. Kaiser,

S. Gouws, Y. Kato, T. Kudo, H. Kazawa and J. Dean, “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”, , 09 2016.

8. Hassan, H., A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li, S. Liu, T.-Y. Liu, R. Luo, A. Menezes, T. Qin, F. Seide, X. Tan, F. Tian, L. Wu, S. Wu, Y. Xia, D. Zhang, Z. Zhang and M. Zhou, “Achieving Human Parity on Automatic Chinese to English News Translation”, *ArXiv*, Vol. abs/1803.05567, 2018.

9. Forcada, M. L. and R. P. Ñeco, “Recursive hetero-associative memories for translation”, J. Mira, R. Moreno-Díaz and J. Cabestany (Editors), *Biological and Artificial Computation: From Neuroscience to Technology*, pp. 453–462, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

10. Castaño, M. A., F. Casacuberta and E. Vidal, “Machine Translation using Neural Networks and Finite-State Models”, pp. 160–167, 1997.

11. Bengio, Y., R. Ducharme, P. Vincent and C. Janvin, “A Neural Probabilistic Language Model”, *J. Mach. Learn. Res.*, Vol. 3, No. null, p. 1137–1155, Mar. 2003.

12. Zamora-Martínez, F., M. J. Bleda and H. Schwenk, “N-gram-based machine translation enhanced with neural networks for the French-English BTEC-IWSLT’10 task”, *IWSLT*, 2010.

13. Schwenk, H., “Continuous Space Translation Models for Phrase-Based Statistical Machine Translation”, *Proceedings of COLING 2012: Posters*, pp. 1071–1080, The COLING 2012 Organizing Committee, Mumbai, India, Dec. 2012, `https://www.aclweb.org/anthology/C12-2104`.

14. Kanouchi, S., K. Sudoh and M. Komachi, “Neural Reordering Model Considering Phrase Translation and Word Alignment for Phrase-based Translation”,

Proceedings of the 3rd Workshop on Asian Translation (WAT2016), pp. 94–103, The COLING 2016 Organizing Committee, Osaka, Japan, Dec. 2016, `https://aclanthology.org/W16-4607`.

15. de Gispert, A., G. Iglesias and B. Byrne, “Fast and Accurate Preordering for SMT using Neural Networks”, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1012–1017, Association for Computational Linguistics, Denver, Colorado, May–Jun. 2015, `https://aclanthology.org/N15-1105`.

16. Devlin, J., R. Zbib, Z. Huang, T. Lamar, R. Schwartz and J. Makhoul, “Fast and Robust Neural Network Joint Models for Statistical Machine Translation”, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1370–1380, Association for Computational Linguistics, Baltimore, Maryland, Jun. 2014, `https://www.aclweb.org/anthology/P14-1129`.

17. Stahlberg, F., “Neural Machine Translation: A Review”, *CoRR*, Vol. abs/1912.02047, 2019, `http://arxiv.org/abs/1912.02047`.

18. Kalchbrenner, N. and P. Blunsom, “Recurrent Continuous Translation Models”, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, Association for Computational Linguistics, Seattle, Washington, USA, Oct. 2013, `https://www.aclweb.org/anthology/D13-1176`.

19. Cho, K., B. van Merriënboer, D. Bahdanau and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches”, *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Association for Computational Linguistics, Doha, Qatar, Oct. 2014, `https://aclanthology.org/W14-4012`.

20. Sutskever, I., O. Vinyals and Q. V. Le, “Sequence to Sequence Learning with

Neural Networks", *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, p. 3104–3112, MIT Press, Cambridge, MA, USA, 2014.

21. Luong, T., H. Pham and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Association for Computational Linguistics, Lisbon, Portugal, Sep. 2015, `https://www.aclweb.org/anthology/D15-1166`.

22. Liu, L., M. Utiyama, A. Finch and E. Sumita, "Agreement on Target-bidirectional Neural Machine Translation", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 411–416, Association for Computational Linguistics, San Diego, California, Jun. 2016, `https://aclanthology.org/N16-1046`.

23. Sennrich, R., B. Haddow and A. Birch, "Edinburgh Neural Machine Translation Systems for WMT 16", *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 371–376, Association for Computational Linguistics, Berlin, Germany, Aug. 2016, `https://aclanthology.org/W16-2323`.

24. Hoang, C. D. V., G. Haffari and T. Cohn, "Towards Decoding as Continuous Optimisation in Neural Machine Translation", *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 146–156, Association for Computational Linguistics, Copenhagen, Denmark, Sep. 2017, `https://aclanthology.org/D17-1014`.

25. Zhang, X., J. Su, Y. Qin, Y. Liu, R. Ji and H. Wang, "Asynchronous Bidirectional Decoding for Neural Machine Translation", *CoRR*, Vol. abs/1801.05122, 2018, `http://arxiv.org/abs/1801.05122`.

26. Zhou, L., J. Zhang and C. Zong, "Synchronous Bidirectional Neural Machine

Translation", *Transactions of the Association for Computational Linguistics*, Vol. 7, pp. 91–105, Mar. 2019, https://aclanthology.org/Q19-1006.

27. Edunov, S., M. Ott, M. Auli and D. Grangier, "Understanding Back-Translation at Scale", *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018, https://aclanthology.org/D18-1045.

28. Chen, M. X., O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu and M. Hughes, "The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation", *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 76–86, Association for Computational Linguistics, Melbourne, Australia, Jul. 2018, https://aclanthology.org/P18-1008.

29. Bapna, A., M. Chen, O. Firat, Y. Cao and Y. Wu, "Training Deeper Neural Machine Translation Models with Transparent Attention", *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3028–3033, Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018, https://aclanthology.org/D18-1338.

30. Wang, Q., B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong and L. S. Chao, "Learning Deep Transformer Models for Machine Translation", *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1810–1822, Association for Computational Linguistics, Florence, Italy, Jul. 2019, https://aclanthology.org/P19-1176.

31. So, D. R., C. Liang and Q. V. Le, "The Evolved Transformer", *CoRR*, Vol. abs/1901.11117, 2019, http://arxiv.org/abs/1901.11117.

32. Sennrich, R., O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-

Dowmunt, S. Läubli, A. V. Miceli Barone, J. Mokry and M. Nădejde, “Nematus: a Toolkit for Neural Machine Translation”, *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 65–68, Association for Computational Linguistics, Valencia, Spain, Apr. 2017, `https://aclanthology.org/E17-3017`.

33. Klein, G., Y. Kim, Y. Deng, J. Senellart and A. Rush, “OpenNMT: Open-Source Toolkit for Neural Machine Translation”, *Proceedings of ACL 2017, System Demonstrations*, pp. 67–72, Association for Computational Linguistics, Vancouver, Canada, Jul. 2017, `https://aclanthology.org/P17-4012`.

34. Vaswani, A., S. Bengio, E. Brevdo, F. Chollet, A. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer and J. Uszkoreit, “Tensor2Tensor for Neural Machine Translation”, *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pp. 193–199, Association for Machine Translation in the Americas, Boston, MA, Mar. 2018, `https://aclanthology.org/W18-1819`.

35. Ott, M., S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier and M. Auli, “fairseq: A Fast, Extensible Toolkit for Sequence Modeling”, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, Association for Computational Linguistics, Minneapolis, Minnesota, Jun. 2019, `https://aclanthology.org/N19-4009`.

36. Junczys-Dowmunt, M., R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins and A. Birch, “Marian: Fast Neural Machine Translation in C++”, *Proceedings of ACL 2018, System Demonstrations*, pp. 116–121, Association for Computational Linguistics, Melbourne, Australia, July 2018, `http://www.aclweb.org/anthology/P18-4020`.

37. Çaglar Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk and Y. Bengio, "On Using Monolingual Corpora in Neural Machine Translation", *ArXiv*, Vol. abs/1503.03535, 2015.

38. Jean, S., O. Firat, K. Cho, R. Memisevic and Y. Bengio, "Montreal Neural Machine Translation Systems for WMT'15", *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 134–140, Association for Computational Linguistics, Lisbon, Portugal, Sep. 2015, `https://aclanthology.org/W15-3014`.

39. Sennrich, R., B. Haddow and A. Birch, "Improving Neural Machine Translation Models with Monolingual Data", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, Association for Computational Linguistics, Berlin, Germany, Aug. 2016, `https://aclanthology.org/P16-1009`.

40. Poncelas, A., D. Shterionov, A. Way, G. M. D. B. Wenniger and P. Passban, "Investigating Backtranslation in Neural Machine Translation", *ArXiv*, Vol. abs/1804.06189, 2018.

41. Hoang, V. C. D., P. Koehn, G. Haffari and T. Cohn, "Iterative Back-Translation for Neural Machine Translation", *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 18–24, Association for Computational Linguistics, Melbourne, Australia, Jul. 2018, `https://aclanthology.org/W18-2703`.

42. Luo, G.-X., Y. Yang, R. Dong, Y.-H. Chen and W. Zhang, "A Joint Back-Translation and Transfer Learning Method for Low-Resource Neural Machine Translation", *Mathematical Problems in Engineering*, Vol. 2020, pp. 1–11, 2020.

43. Imamura, K., A. Fujita and E. Sumita, "Enhancement of Encoder and Attention Using Target Monolingual Corpora in Neural Machine Translation", *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 55–63, Association for Computational Linguistics, Melbourne, Australia, Jul. 2018,

https://aclanthology.org/W18-2707.

44. Caswell, I., C. Chelba and D. Grangier, “Tagged Back-Translation”, *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pp. 53–63, Association for Computational Linguistics, Florence, Italy, Aug. 2019, https://aclanthology.org/W19-5206.

45. Wang, S., Y. Liu, C. Wang, H. Luan and M. Sun, “Improving Back-Translation with Uncertainty-based Confidence Estimation”, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 791–802, Association for Computational Linguistics, Hong Kong, China, Nov. 2019, https://aclanthology.org/D19-1073.

46. Currey, A., A. V. Miceli Barone and K. Heafield, “Copied Monolingual Data Improves Low-Resource Neural Machine Translation”, *Proceedings of the Second Conference on Machine Translation*, pp. 148–156, Association for Computational Linguistics, Copenhagen, Denmark, Sep. 2017, https://aclanthology.org/W17-4715.

47. Zhang, J. and C. Zong, “Exploiting Source-side Monolingual Data in Neural Machine Translation”, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1535–1545, Association for Computational Linguistics, Austin, Texas, Nov. 2016, https://aclanthology.org/D16-1160.

48. He, J., J. Gu, J. Shen and M. Ranzato, “Revisiting Self-Training for Neural Sequence Generation”, *CoRR*, Vol. abs/1909.13788, 2019, http://arxiv.org/abs/1909.13788.

49. Jiao, W., X. Wang, Z. Tu, S. Shi, M. R. Lyu and I. King, “Self-Training Sampling with Monolingual Data Uncertainty for Neural Machine Translation”, *CoRR*, Vol. abs/2106.00941, 2021, https://arxiv.org/abs/2106.00941.

50. Wu, L., Y. Wang, Y. Xia, T. Qin, J. Lai and T.-Y. Liu, "Exploiting Monolingual Data at Scale for Neural Machine Translation", *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4207–4216, Association for Computational Linguistics, Hong Kong, China, Nov. 2019, `https://aclanthology.org/D19-1430`.

51. Cheng, Y., W. Xu, Z. He, W. He, H. Wu, M. Sun and Y. Liu, "Semi-Supervised Learning for Neural Machine Translation", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1965–1974, Association for Computational Linguistics, Berlin, Germany, Aug. 2016, `https://aclanthology.org/P16-1185`.

52. He, D., Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu and W.-Y. Ma, "Dual Learning for Machine Translation", *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, p. 820–828, Curran Associates Inc., Red Hook, NY, USA, 2016.

53. Zheng, Z., H. Zhou, S. Huang, L. Li, X.-Y. Dai and J. Chen, "Mirror-Generative Neural Machine Translation", *International Conference on Learning Representations*, 2020, `https://openreview.net/forum?id=HkxQRTNYPH`.

54. Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep Contextualized Word Representations", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, Association for Computational Linguistics, New Orleans, Louisiana, Jun. 2018, `https://aclanthology.org/N18-1202`.

55. Devlin, J., M. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *CoRR*, Vol. abs/1810.04805, 2018, `http://arxiv.org/abs/1810.04805`.

56. Edunov, S., A. Baevski and M. Auli, “Pre-trained language model representations for language generation”, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4052–4059, Association for Computational Linguistics, Minneapolis, Minnesota, Jun. 2019, `https://aclanthology.org/N19-1409`.

57. Song, K., X. Tan, T. Qin, J. Lu and T. Liu, “MASS: Masked Sequence to Sequence Pre-training for Language Generation”, *CoRR*, Vol. abs/1905.02450, 2019, `http://arxiv.org/abs/1905.02450`.

58. Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Association for Computational Linguistics, Online, Jul. 2020, `https://aclanthology.org/2020.acl-main.703`.

59. Liu, Y., J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis and L. Zettlemoyer, “Multilingual Denoising Pre-training for Neural Machine Translation”, *CoRR*, Vol. abs/2001.08210, 2020, `https://arxiv.org/abs/2001.08210`.

60. Fadaee, M., A. Bisazza and C. Monz, “Data Augmentation for Low-Resource Neural Machine Translation”, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 567–573, Association for Computational Linguistics, Vancouver, Canada, Jul. 2017, `https://aclanthology.org/P17-2090`.

61. Sennrich, R. and B. Zhang, “Revisiting Low-Resource Neural Machine Translation: A Case Study”, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 211–221, Association for Computational Linguistics, Florence, Italy, Jul. 2019, `https://aclanthology.org/P19-1021`.

62. Bojar, O., R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia and M. Turchi, "Findings of the 2017 Conference on Machine Translation (WMT17)", *Proceedings of the Second Conference on Machine Translation*, pp. 169–214, Association for Computational Linguistics, Copenhagen, Denmark, Sep. 2017, https://aclanthology.org/W17-4717.

63. García-Martínez, M., O. Caglayan, W. Aransa, A. Bardet, F. Bougares and L. Barrault, "LIUM Machine Translation Systems for WMT17 News Translation Task", *Proceedings of the Second Conference on Machine Translation*, pp. 288–295, Association for Computational Linguistics, Copenhagen, Denmark, Sep. 2017, https://aclanthology.org/W17-4726.

64. Gwinnup, J., T. Anderson, G. Erdmann, K. Young, M. Kazi, E. Salesky, B. Thompson and J. Taylor, "The AFRL-MITLL WMT17 Systems: Old, New, Borrowed, BLEU", *Proceedings of the Second Conference on Machine Translation*, pp. 303–309, Association for Computational Linguistics, Copenhagen, Denmark, Sep. 2017, https://aclanthology.org/W17-4728.

65. Sennrich, R., A. Birch, A. Currey, U. Germann, B. Haddow, K. Heafield, A. V. Miceli Barone and P. Williams, "The University of Edinburgh's Neural MT Systems for WMT17", *Proceedings of the Second Conference on Machine Translation*, pp. 389–399, Association for Computational Linguistics, Copenhagen, Denmark, Sep. 2017, https://aclanthology.org/W17-4739.

66. Zhou, J., Y. Cao, X. Wang, P. Li and W. Xu, "Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation", *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 371–383, 2016.

67. Bojar, O., C. Federmann, M. Fishel, Y. Graham, B. Haddow, P. Koehn and C. Monz, "Findings of the 2018 Conference on Machine Translation (WMT18)", *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*,

pp. 272–303, Association for Computational Linguistics, Belgium, Brussels, Oct. 2018, https://aclanthology.org/W18-6401.

68. Marie, B., R. Wang, A. Fujita, M. Utiyama and E. Sumita, "NICT's Neural and Statistical Machine Translation Systems for the WMT18 News Translation Task", *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 449–455, Association for Computational Linguistics, Belgium, Brussels, Oct. 2018, https://aclanthology.org/W18-6419.

69. Haddow, B., N. Bogoychev, D. Emelin, U. Germann, R. Grundkiewicz, K. Heafield, A. V. Miceli Barone and R. Sennrich, "The University of Edinburgh's Submissions to the WMT18 News Translation Task", *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 399–409, Association for Computational Linguistics, Belgium, Brussels, Oct. 2018, https://aclanthology.org/W18-6412.

70. Sennrich, R., B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Association for Computational Linguistics, Berlin, Germany, Aug. 2016, https://aclanthology.org/P16-1162.

71. Rsennrich, "rsennrich/subword-nmt", https://github.com/rsennrich/subword-nmt.

72. Jean, S., K. Cho, R. Memisevic and Y. Bengio, "On Using Very Large Target Vocabulary for Neural Machine Translation", *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)*, pp. 1–10, Association for Computational Linguistics, Beijing, China, Jul. 2015, https://aclanthology.org/P15-1001.

73. Luong, T., I. Sutskever, Q. Le, O. Vinyals and W. Zaremba, “Addressing the Rare Word Problem in Neural Machine Translation”, *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)*, pp. 11–19, Association for Computational Linguistics, Beijing, China, Jul. 2015, `https://aclanthology.org/P15-1002`.

74. Luong, M.-T. and C. D. Manning, “Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models”, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1054–1063, Association for Computational Linguistics, Berlin, Germany, Aug. 2016, `https://aclanthology.org/P16-1100`.

75. Sak, H., T. Güngör and M. Saraçlar, “Morphological Disambiguation of Turkish Text with Perceptron Algorithm”, A. Gelbukh (Editor), *Computational Linguistics and Intelligent Text Processing*, pp. 107–118, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

76. Shen, Y., X. Tan, D. He, T. Qin and T.-Y. Liu, “Dense Information Flow for Neural Machine Translation”, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1294–1303, Association for Computational Linguistics, New Orleans, Louisiana, Jun. 2018, `https://aclanthology.org/N18-1117`.

77. Bektaş, E., E. Yilmaz, C. Mermer and İ. Durgar El-Kahlout, “TÜBİTAK SMT System Submission for WMT2016”, *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 246–251, Association for Computational Linguistics, Berlin, Germany, Aug. 2016, `https://aclanthology.org/W16-2305`.

78. Oflazer, K., “Two-level Description of Turkish Morphology”, *Sixth Conference*

of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Utrecht, The Netherlands, Apr. 1993, https://aclanthology.org/E93-1066.

79. Ataman, D., M. Negri, M. Turchi and M. Federico, "Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English", *The Prague Bulletin of Mathematical Linguistics*, Vol. 108, 06 2017.

80. Pan, Y., X. Li, Y. Yang and R. Dong, "Multi-Source Neural Model for Machine Translation of Agglutinative Language", *Future Internet*, Vol. 12, p. 96, 06 2020.

81. Tyers, F. M. and M. S. Alperen, "South-East European Times: A parallel corpus of Balkan languages", *Proceedings of the LREC Workshop on Exploitation of Multilingual Resources and Tools for Central and (South-) Eastern European Languages*, Valetta, Malta, 2010.

82. Tiedemann, J., "Parallel Data, Tools and Interfaces in OPUS", N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk and S. Piperidis (Editors), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA), Istanbul, Turkey, 05 2012.

83. Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation", *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp. 177–180, Association for Computational Linguistics, Prague, Czech Republic, Jun. 2007, https://aclanthology.org/P07-2045.

84. Barrault, L., M. Biesialska, O. Bojar, M. R. Costa-jussà, C. Federmann, Y. Graham, R. Grundkiewicz, B. Haddow, M. Huck, E. Joanis, T. Kocmi, P. Koehn,

C.-k. Lo, N. Ljubešić, C. Monz, M. Morishita, M. Nagata, T. Nakazawa, S. Pal, M. Post and M. Zampieri, "Findings of the 2020 Conference on Machine Translation (WMT20)", *Proceedings of the Fifth Conference on Machine Translation*, pp. 1–55, Association for Computational Linguistics, Online, Nov. 2020, `https://aclanthology.org/2020.wmt-1.1`.

85. Pascanu, R., Çaglar Gülçehre, K. Cho and Y. Bengio, "How to Construct Deep Recurrent Neural Networks", *CoRR*, Vol. abs/1312.6026, 2014.

86. Sennrich, R. and B. Haddow, "Linguistic Input Features Improve Neural Machine Translation", *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pp. 83–91, Association for Computational Linguistics, Berlin, Germany, Aug. 2016, `https://aclanthology.org/W16-2209`.

87. Junczys-Dowmunt, M. and R. Grundkiewicz, "An Exploration of Neural Sequence-to-Sequence Architectures for Automatic Post-Editing", *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 120–129, Asian Federation of Natural Language Processing, Taipei, Taiwan, Nov. 2017, `https://aclanthology.org/I17-1013`.

88. Press, O. and L. Wolf, "Using the Output Embedding to Improve Language Models", *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, Association for Computational Linguistics, Valencia, Spain, Apr. 2017, `https://aclanthology.org/E17-2025`.

89. Ba, J., J. Kiros and G. E. Hinton, "Layer Normalization", *ArXiv*, Vol. abs/1607.06450, 2016.

90. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", Y. Bengio and Y. LeCun (Editors), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Pro-*

ceedings, 2015, `http://arxiv.org/abs/1412.6980`.

91. Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision”, *CoRR*, Vol. abs/1512.00567, 2015, `http://arxiv.org/abs/1512.00567`.

92. Durgar El-Kahlout, İ., E. Bektaş, N. Ş. Erdem and H. Kaya, “Translating Between Morphologically Rich Languages: An Arabic-to-Turkish Machine Translation System”, *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 158–166, Association for Computational Linguistics, Florence, Italy, Aug. 2019, `https://aclanthology.org/W19-4617`.

93. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest and A. M. Rush, “HuggingFace’s Transformers: State-of-the-art Natural Language Processing”, *ArXiv*, Vol. abs/1910.03771, 2019.

94. Schweter, S., “BERTurk - BERT models for Turkish”, , Apr. 2020, `https://doi.org/10.5281/zenodo.3770924`.

95. Sanh, V., L. Debut, J. Chaumond and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”, *CoRR*, Vol. abs/1910.01108, 2019, `http://arxiv.org/abs/1910.01108`.

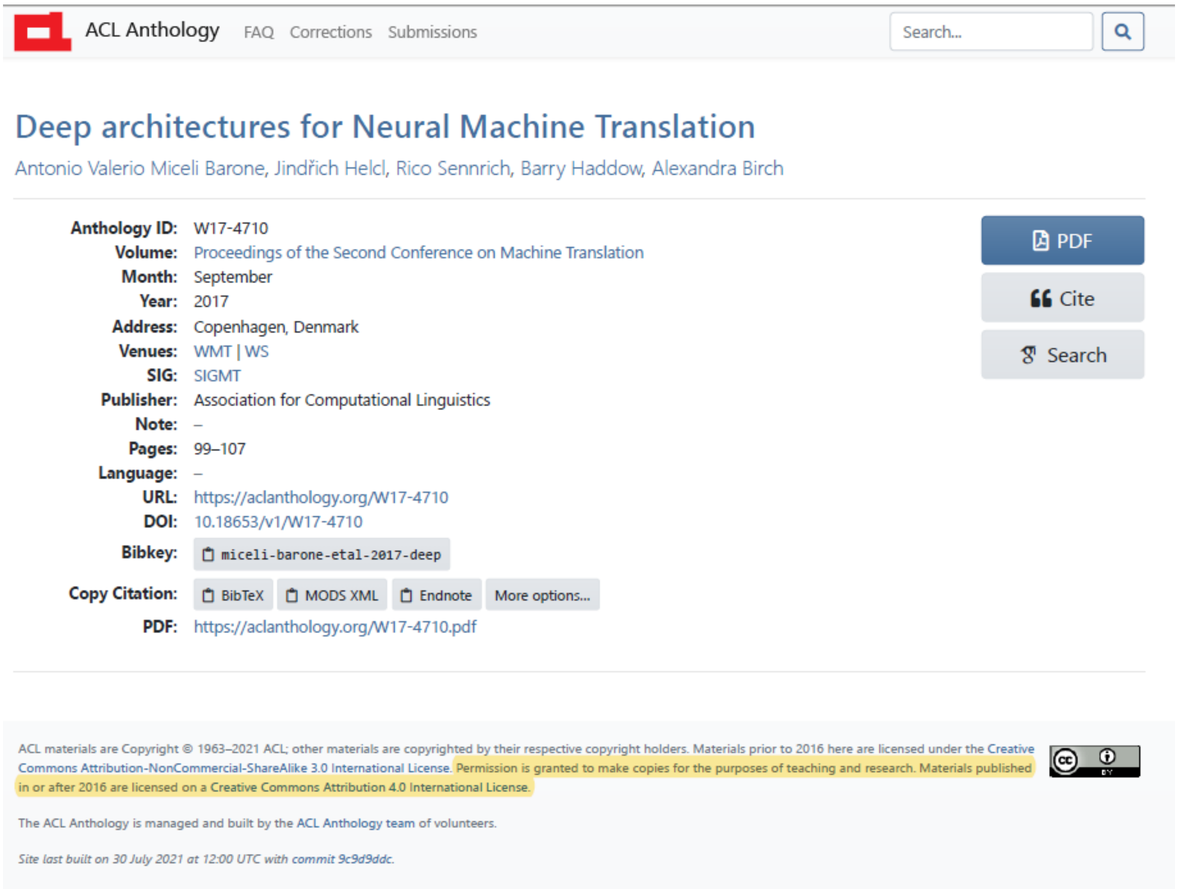
96. Oflazer, K., B. Say, D. Zeynep and G. Tur, “Building a Turkish Treebank”, *Abeillé*, 01 2003.

97. Imamura, K. and E. Sumita, “Ensemble and Reranking: Using Multiple Models in the NICT-2 Neural Machine Translation System at WAT2017”, *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pp. 127–134, Asian Federation of Natural Language Processing, Taipei, Taiwan, Nov. 2017,

https://aclanthology.org/W17-5711.

APPENDIX A: COPYRIGHT PERMISSION FOR FIGURES 4.3 AND 4.5

Copyright license of Figures 4.3 and 4.5 are given in Figure A.1. As shown in the figure, the article ”Deep Architectures for Neural Machine Translation” published on Association for Computational Linguistics (ACL) is licensed on a Creative Commons Attribution 4.0 International License (Figure A.2). The link of the license is provided as suggested in the license description for the reuse of images: https://creativecommons.org/licenses/by/4.0/legalcode.



Deep architectures for Neural Machine Translation

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, Alexandra Birch

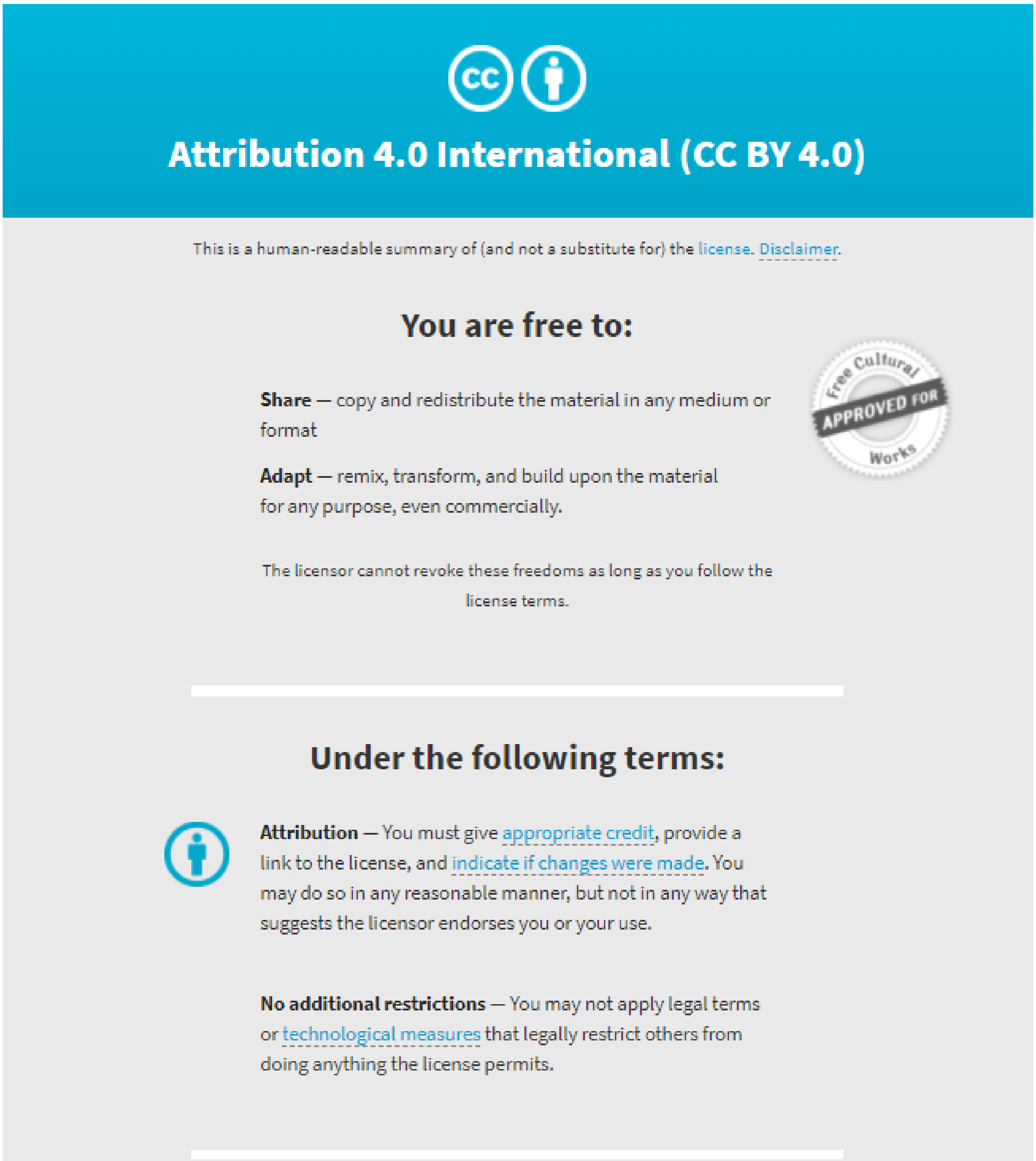
Anthology ID: W17-4710
Volume: Proceedings of the Second Conference on Machine Translation
Month: September
Year: 2017
Address: Copenhagen, Denmark
Venues: WMT | WS
SIG: SIGMT
Publisher: Association for Computational Linguistics
Note: –
Pages: 99–107
Language: –
URL: https://aclanthology.org/W17-4710
DOI: 10.18653/v1/W17-4710
Bibkey: `miceli-barone-etal-2017-deep`
Copy Citation: BibTeX MODS XML Endnote More options...
PDF: https://aclanthology.org/W17-4710.pdf

ACL materials are Copyright © 1963–2021 ACL; other materials are copyrighted by their respective copyright holders. Materials prior to 2016 here are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License. Permission is granted to make copies for the purposes of teaching and research. Materials published in or after 2016 are licensed on a Creative Commons Attribution 4.0 International License.

The ACL Anthology is managed and built by the ACL Anthology team of volunteers.

Site last built on 30 July 2021 at 12:00 UTC with commit 9c9d9ddc.

Figure A.1. Copyright license of Figures 4.3 and 4.5



Attribution 4.0 International (CC BY 4.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

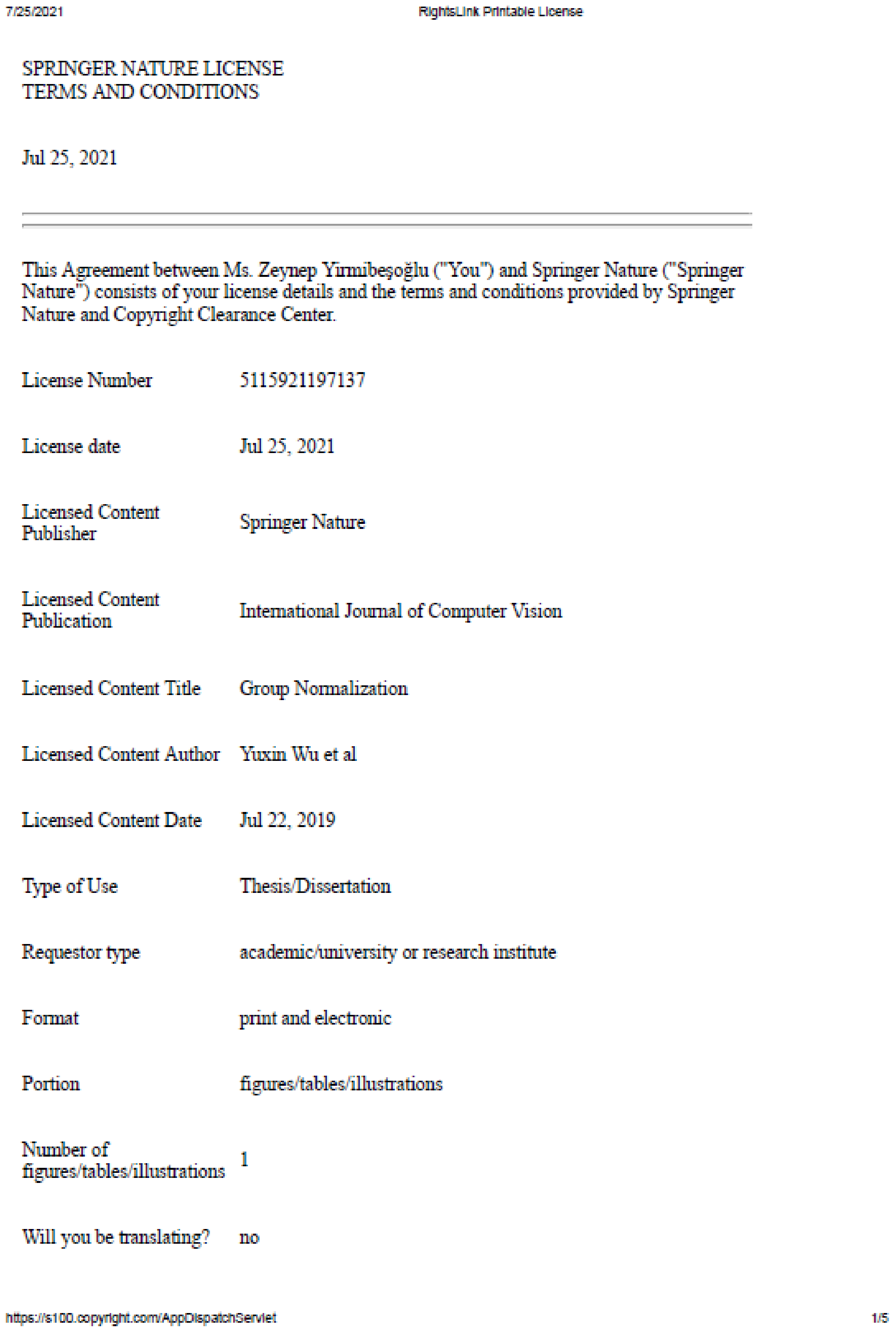
No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Figure A.2. Creative Commons Attribution 4.0 International License

APPENDIX B: COPYRIGHT PERMISSION FOR FIGURES 4.4

Figure 4.4 has been reprinted by permission from RightsLink: Springer, International Journal of Computer Vision, ”Group Normalization”, Wu and He (2018), License number: 5115921197137, 2021.

Copyright license for the reuse of Figure 4.4 is provided in Figures B.1, B.2, B.3, B.4, B.5.



7/25/2021 RightsLink Printable License

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

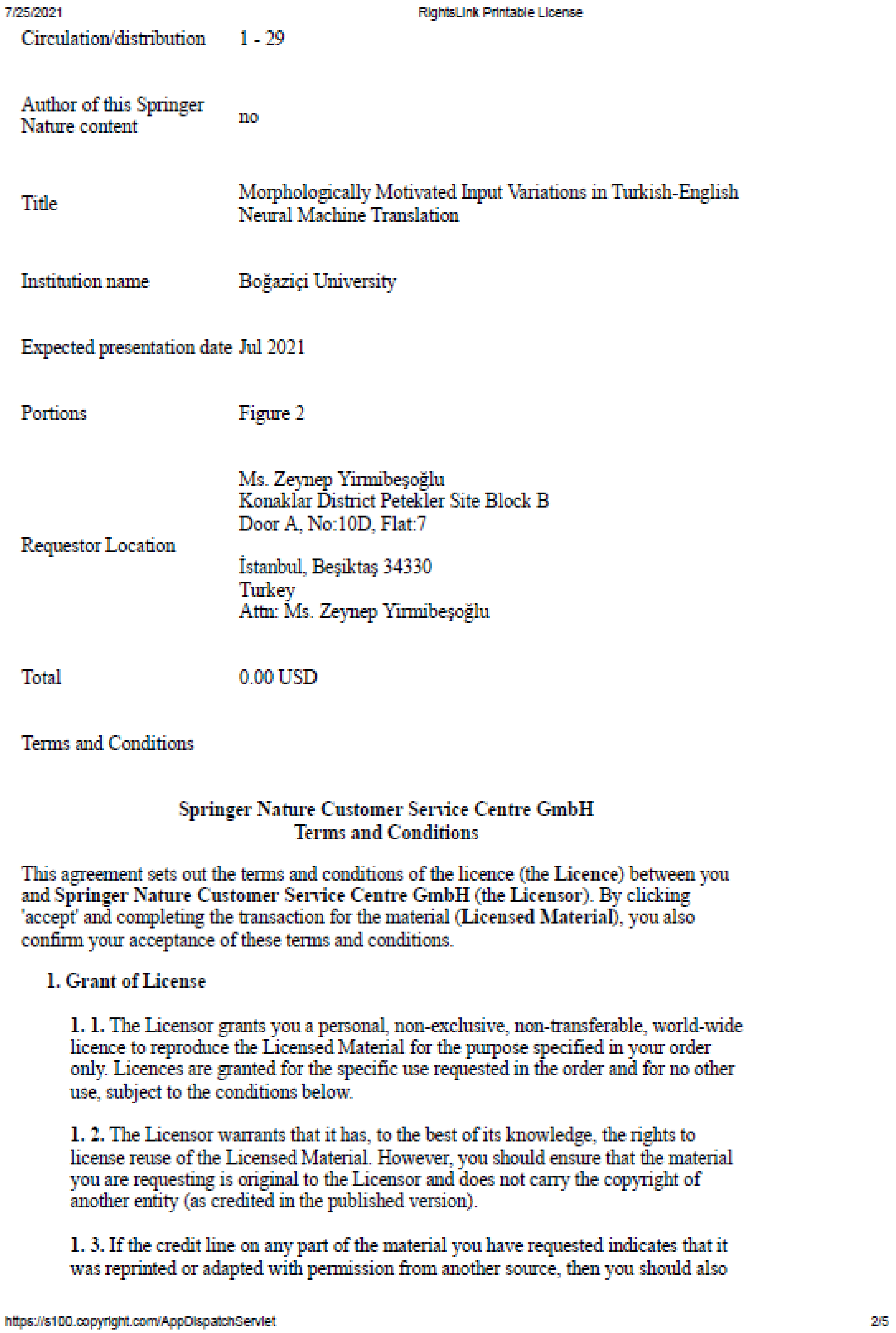
Jul 25, 2021

This Agreement between Ms. Zeynep Yirmibeşoğlu ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5115921197137
License date	Jul 25, 2021
Licensed Content Publisher	Springer Nature
Licensed Content Publication	International Journal of Computer Vision
Licensed Content Title	Group Normalization
Licensed Content Author	Yuxin Wu et al
Licensed Content Date	Jul 22, 2019
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Will you be translating?	no

https://s100.copyright.com/AppDispatchServlet 1/5

Figure B.1. Copyright license for the reuse of Figure 4.4, Page 1



7/25/2021 RightsLink Printable License

Circulation/distribution	1 - 29
Author of this Springer Nature content	no
Title	Morphologically Motivated Input Variations in Turkish-English Neural Machine Translation
Institution name	Boğaziçi University
Expected presentation date	Jul 2021
Portions	Figure 2
Requestor Location	Ms. Zeynep Yirmibeşoğlu Konaklar District Petekler Site Block B Door A, No:10D, Flat:7 İstanbul, Beşiktaş 34330 Turkey Attn: Ms. Zeynep Yirmibeşoğlu
Total	0.00 USD

Terms and Conditions

Springer Nature Customer Service Centre GmbH
Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

1. **Grant of License**

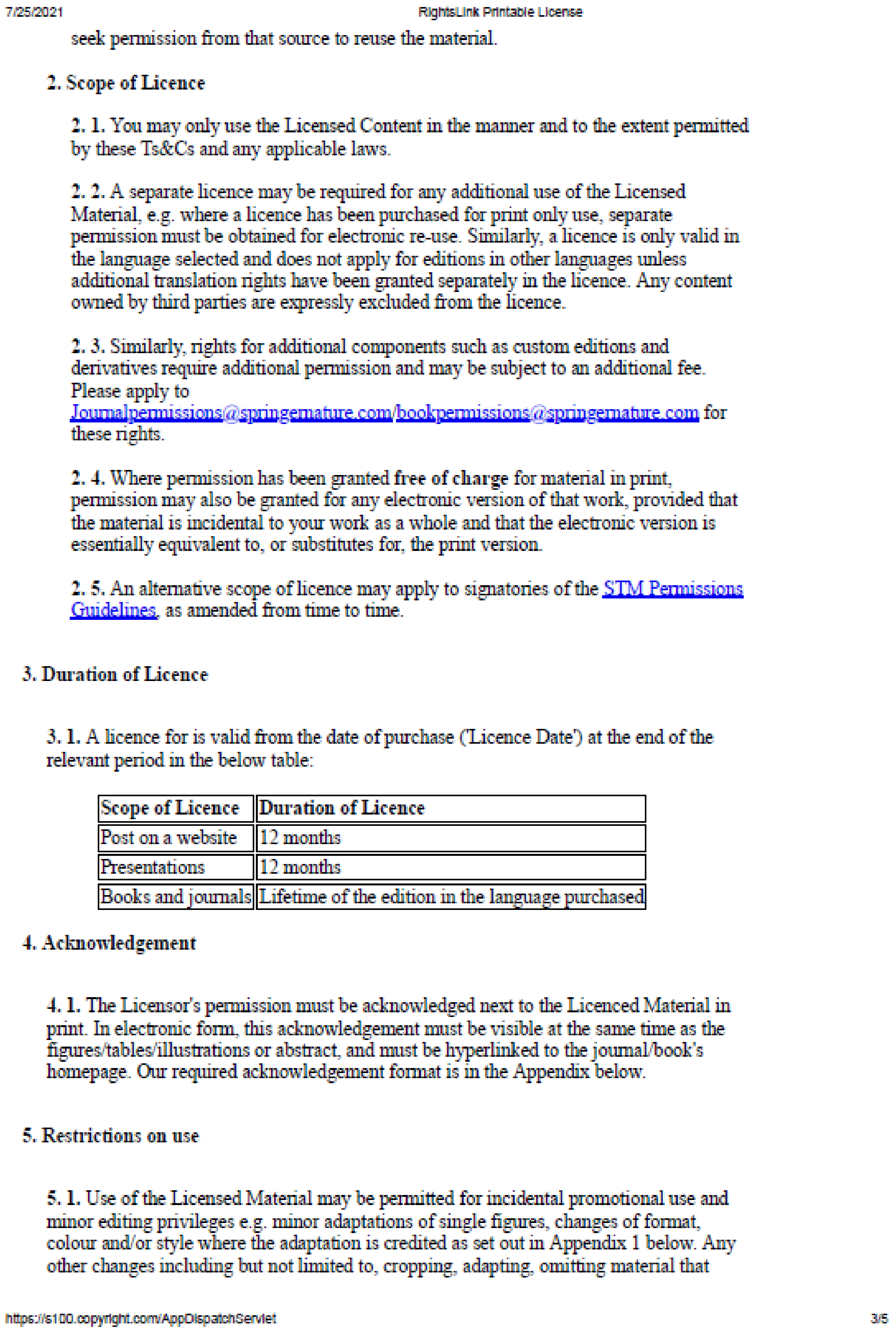
1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

1. 2. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

1. 3. If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also

https://s100.copyright.com/AppDispatchServlet 2/5

Figure B.2. Copyright license for the reuse of Figure 4.4, Page 2



seek permission from that source to reuse the material.

2. Scope of Licence

2. 1. You may only use the Licensed Content in the manner and to the extent permitted by these Ts&Cs and any applicable laws.

2. 2. A separate licence may be required for any additional use of the Licensed Material, e.g. where a licence has been purchased for print only use, separate permission must be obtained for electronic re-use. Similarly, a licence is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence. Any content owned by third parties are expressly excluded from the licence.

2. 3. Similarly, rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

2. 4. Where permission has been granted **free of charge** for material in print, permission may also be granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

2. 5. An alternative scope of licence may apply to signatories of the STM Permissions Guidelines, as amended from time to time.

3. Duration of Licence

3. 1. A licence for is valid from the date of purchase ('Licence Date') at the end of the relevant period in the below table:

Scope of Licence	Duration of Licence
Post on a website	12 months
Presentations	12 months
Books and journals	Lifetime of the edition in the language purchased

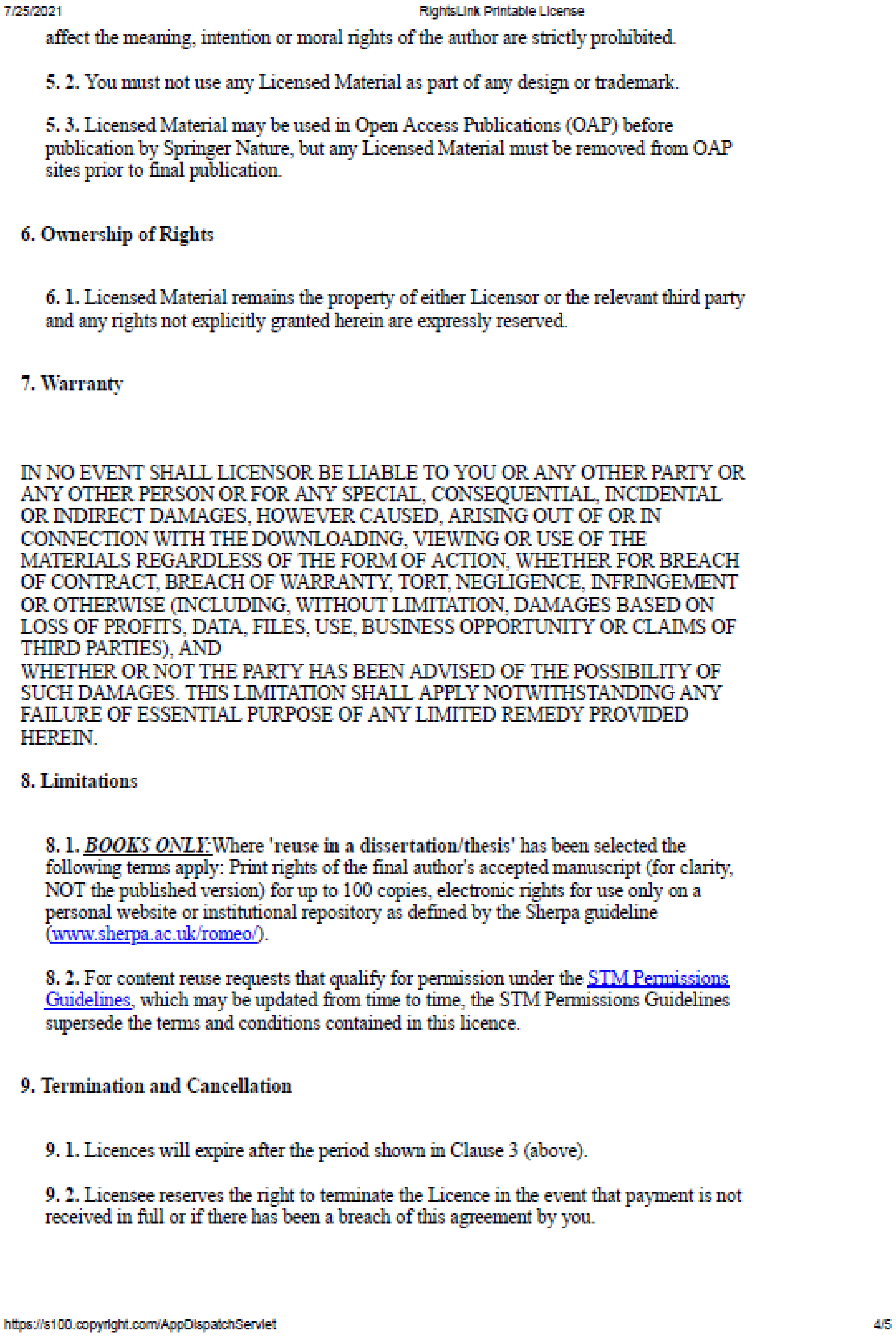
4. Acknowledgement

4. 1. The Licensor's permission must be acknowledged next to the Licenced Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

5. Restrictions on use

5. 1. Use of the Licensed Material may be permitted for incidental promotional use and minor editing privileges e.g. minor adaptations of single figures, changes of format, colour and/or style where the adaptation is credited as set out in Appendix 1 below. Any other changes including but not limited to, cropping, adapting, omitting material that

Figure B.3. Copyright license for the reuse of Figure 4.4, Page 3



affect the meaning, intention or moral rights of the author are strictly prohibited.

5. 2. You must not use any Licensed Material as part of any design or trademark.

5. 3. Licensed Material may be used in Open Access Publications (OAP) before publication by Springer Nature, but any Licensed Material must be removed from OAP sites prior to final publication.

6. Ownership of Rights

6. 1. Licensed Material remains the property of either Licensor or the relevant third party and any rights not explicitly granted herein are expressly reserved.

7. Warranty

IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND
WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

8. Limitations

8. 1. ***BOOKS ONLY:*** Where **'reuse in a dissertation/thesis'** has been selected the following terms apply: Print rights of the final author's accepted manuscript (for clarity, NOT the published version) for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

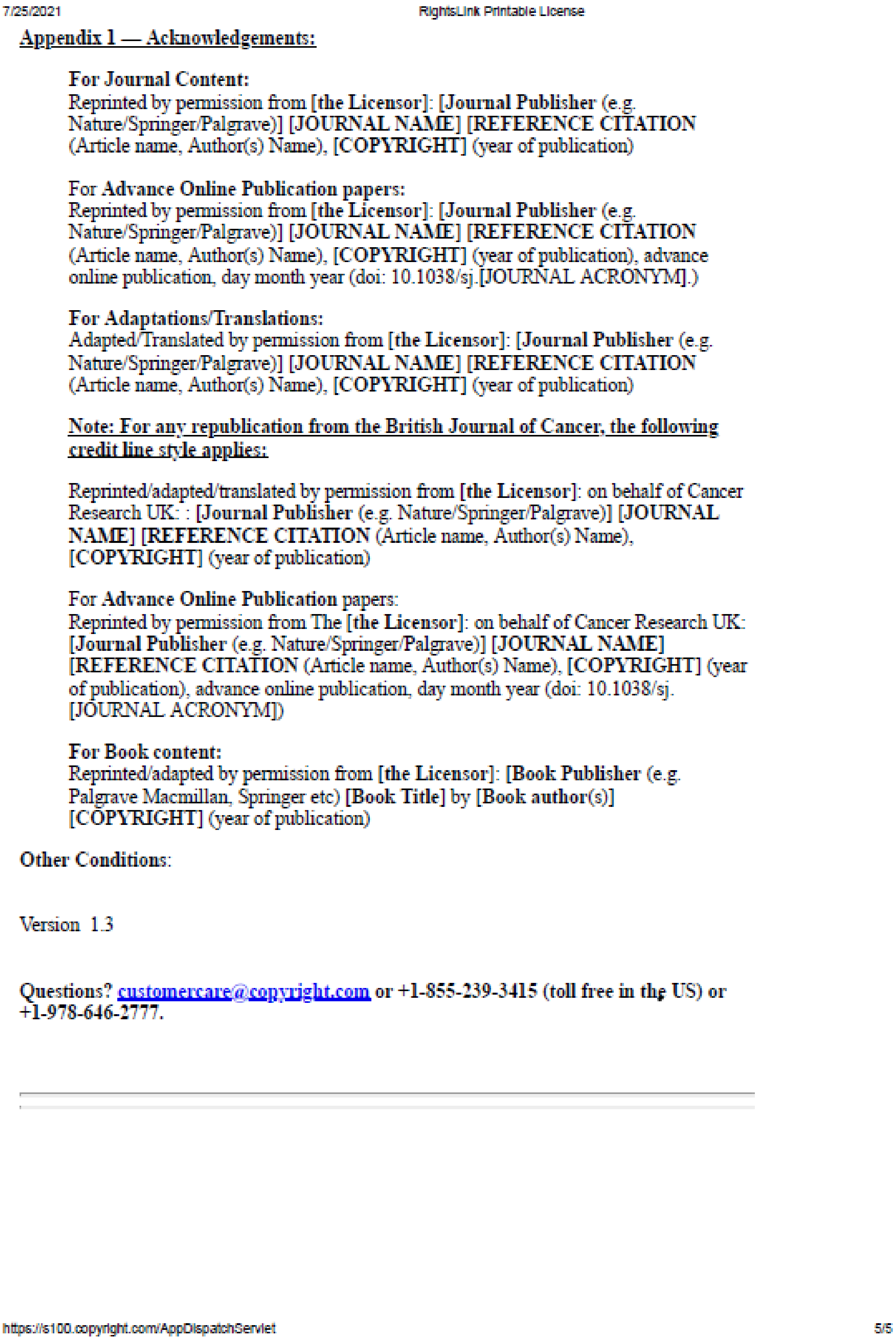
8. 2. For content reuse requests that qualify for permission under the STM Permissions Guidelines, which may be updated from time to time, the STM Permissions Guidelines supersede the terms and conditions contained in this licence.

9. Termination and Cancellation

9. 1. Licences will expire after the period shown in Clause 3 (above).

9. 2. Licensee reserves the right to terminate the Licence in the event that payment is not received in full or if there has been a breach of this agreement by you.

Figure B.4. Copyright license for the reuse of Figure 4.4, Page 4



Appendix 1 — Acknowledgements:

For Journal Content:
Reprinted by permission from **[the Licensor]**: **[Journal Publisher** (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

For **Advance Online Publication papers:**
Reprinted by permission from **[the Licensor]**: **[Journal Publisher** (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)

For Adaptations/Translations:
Adapted/Translated by permission from **[the Licensor]**: **[Journal Publisher** (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from **[the Licensor]**: on behalf of Cancer Research UK: : **[Journal Publisher** (e.g. Nature/Springer/Palgrave)] **[JOURNAL NAME] [REFERENCE CITATION** (Article name, Author(s) Name), **[COPYRIGHT]** (year of publication)

For **Advance Online Publication** papers:
Reprinted by permission from The **[the Licensor]**: on behalf of Cancer Research UK: **[Journal Publisher** (e.g. Nature/Springer/Palgrave)] **[JOURNAL NAME] [REFERENCE CITATION** (Article name, Author(s) Name), **[COPYRIGHT]** (year of publication), advance online publication, day month year (doi: 10.1038/sj. [JOURNAL ACRONYM])

For Book content:
Reprinted/adapted by permission from **[the Licensor]**: **[Book Publisher** (e.g. Palgrave Macmillan, Springer etc) **[Book Title]** by **[Book author(s)]** **[COPYRIGHT]** (year of publication)

Other Conditions:

Version 1.3

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Figure B.5. Copyright license for the reuse of Figure 4.4, Page 5