applied sciences

MDPI

Article

Fully Attentional Network for Low-Resource Academic Machine Translation and Post Editing

İlhami Sel * and Davut Hanbay

Department of Computer Engineering, Faculty of Engineering, Inonu University, Malatya 44200, Turkey
* Correspondence: ilhamisel23@gmail.com

Abstract: English is accepted as an academic language in the world. This necessitates the use of English in their academic studies for speakers of other languages. Even when these researchers are competent in the use of the English language, some mistakes may occur while writing an academic article. To solve this problem, academicians tend to use automatic translation programs or get assistance from people with an advanced level of English. This study offers an expert system to enable assistance to the researchers throughout their academic article writing process. In this study, Turkish which is considered among low-resource languages is used as the source language. The proposed model combines the transformer encoder-decoder architecture model with the pre-trained Sci-BERT language model via the shallow fusion method. The model uses a Fully Attentional Network Layer instead of a Feed-Forward Network Layer in the known shallow fusion method. In this way, a higher success rate could be achieved by increasing the attention at the word level. Different metrics were used to evaluate the created model. The model created as a result of the experiments reached 45.1 BLEU and 73.2 METEOR scores. In addition, the proposed model achieved 20.12 and 20.56 scores, respectively, with the zero-shot translation method in the World Machine Translation (2017–2018) test datasets. The proposed method could inspire other low-resource languages to include the language model in the translation system. In this study, a corpus composed entirely of academic sentences is also introduced to be used in the translation system. The corpus consists of 1.2 million parallel sentences. The proposed model and corpus are made available to researchers on our GitHub page.

Keywords: natural language processing; neural machine translation; transformer; fully attentional network; parallel corpus

check for updates

Citation: Sel, İ.; Hanbay, D. Fully Attentional Network for Low-Resource Academic Machine Translation and Post Editing. *Appl. Sci.* **2022**, *12*, 11456. https://doi.org/10.3390/app122211456

Academic Editor: Rafael Valencia-Garcia

Received: 30 September 2022
Accepted: 6 November 2022
Published: 11 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Machine Translation (MT) is based on sequence-to-sequence Machine Learning (ML) that translates from one language to another without human intervention [1]. MT is crucial not only for academics but also for large technology companies (Google Translate, Bing Translator, Yandex Translate, DeepL, etc.). The performance of machine translation heavily relies on the quality of the parallel corpus. The quality and quantity of the corpus directly affect the quality of translation. Translation models (English-French, English-German) with a large number of resources have achieved the quality of human translation [2]. However, the translation quality has not reached that level for the Turkish-English language pair yet due to the lack of adequate studies and the lack of high-quality parallel data. Google Translate (GT) and DeepL translation systems are currently the most widely used translation systems from Turkish to English. Both translation systems use transformer architecture and create very successful translations. However, academic terminology translations are different from everyday spoken language translations. In this respect, the translation systems sometimes mistranslate academic terms (see Table 1).

Table 1. Some terms that the Google Translate (Version: 6.36, Date: 11 May 2022) system translates incorrectly or incompletely.

Turkish in the Literature	Google Translate	In the Literature
evrişimsel sinir ağı girişleri	convolutional neural network entries	Convolutional neural network inputs
sözel olmayan yakınlık becerileri	non-verbal intimacy skills	Non-verbal immediacy skills
nitel araştırmalarda çeşitleme	diversification in qualitative research	Triangulation in qualitative research
sınıf içi öğretmen davranışları	classroom teacher behavior	Teacher behavior within classroom
belirsizlik hoşgörüsü seviyesi	level of uncertainty tolerance	Ambiguity tolerance level

In this study, firstly, a parallel corpus was created with the use of the abstracts of graduate theses in Turkey. Researchers in Turkey are supposed to prepare the abstracts in Turkish and English and researchers' dissertations are uploaded to the Council of Higher Education's website "CoHE's Thesis System" https://tez.yok.gov.tr/ (Accessed on 3 November 2022). The method we use while creating the corpus consists of three stages. First, the abstracts of the theses in "CoHE's Thesis System" were recorded in text files. Secondly, text files were separated into sentences with text pre-processing steps. Finally, sentences were aligned via the developed sentence alignment algorithms to find the English sentence corresponding to each Turkish sentence. 1,217,300 parallel sentences were obtained after sentence alignment algorithms. Secondly, a neural machine translation system was created using the obtained parallel corpus. There are two main transformer architectures in this proposed model. The first architecture is the encoder-decoder network, and the second architecture is a pre-trained language model that has achieved high success in Natural Language Generation tasks. The shallow fusion method was employed to benefit most from these two architectures. In the shallow fusion method, the last decoder network layer output is introduced as the input of the language model. Then, the language model output and the decoder network output are combined via an additional neural network. In the known shallow fusion method Feed-Forward Network (FFN) is used for this network. On the other hand, a Fully Attentional Network (FAN) is used in the proposed model for this network. The proposed architecture is explained in detail in the method section. The corpus of the proposed model was also utilized to develop new models with the current neural machine translation architectures (LSTM, Convolutional, Base Transformer). The success of the proposed model was compared to the performance of the models with current neural machine translation architectures. In addition to the translation systems, the test set was translated sentence by sentence via GT. Metrics BLEU, METEOR, TER, and Perplexity were used to evaluate the models' success. FAN architecture used model scored 45.1 BLEU and 73.2 METEOR score and achieved higher success than FFN architecture.

In addition, the proposed model achieved 20.12 and 20.56 BLEU scores respectively on the World Machine Translation (WMT) (2017–2018) test data sets via the zero-shot translation model.

Problem Definitions and Motivation

Especially in recent years, translation services (Google, Bing, Yahoo) developed by large corporations have made considerable progress [3]. For daily spoken language translations, when the source sentence entered into the system is a smooth and grammatically correct sequential sentence, the target sentence structure will be clear and comprehensible in parallel. The abundance and variation of scientific terms in academic studies may decrease the effectiveness of these translation systems. While there are examples of academic translation systems for other languages such as Aspec [4] and Scielo Corpus [5], there is still no research on the Turkish language. The need for an academic translation system and advances in NLP and MT are factors that motivate this study in its emergence. In this study, both a corpus and an NMT model are proposed for Turkish-English academic translations.

- This study offers a translation system for researchers and graduate students to get better translations while writing academic papers in English. In this way, spelling and grammar errors can be minimized.
- This study introduces a parallel corpus that has been created comprehensively for the Turkish-English language pair.
- The use of the shallow fusion method added to the base of the translation architectures represented in this study may inspire further studies on different language pairs.
- The parallel corpus created and the proposed model are presented as open-source for researchers working in the field of machine translation. https://github.com/ilhamisel/tr-en_translate (Accessed on 3 November 2022).

2. Related Works

The first subsection includes current studies on machine translation. The second section includes the studies that we have reached so far in the field of Turkish English machine translation.

2.1. Machine Translation

Machine translation systems are grouped under 3 main headings: Rule-based, Statistical, and Neural Machine Translation Systems [6]. Even though Rule-based and Statistical MT systems reached a considerable level of translation success, MT has made great progress with the development of Artificial Neural Networks and Deep Learning in the last 10 years. Modern NMT systems emerged with the addition of RNN networks to statistical machine translation systems. Later, deep neural networks and End-to-End translation models emerged [7]. RNN and its variants such as LSTM, BiLSTM, and GRU have progressed greatly as NMT systems. Encoder-decoder is the most traditional architecture used in Neural Machine Translations (see Figure 1).

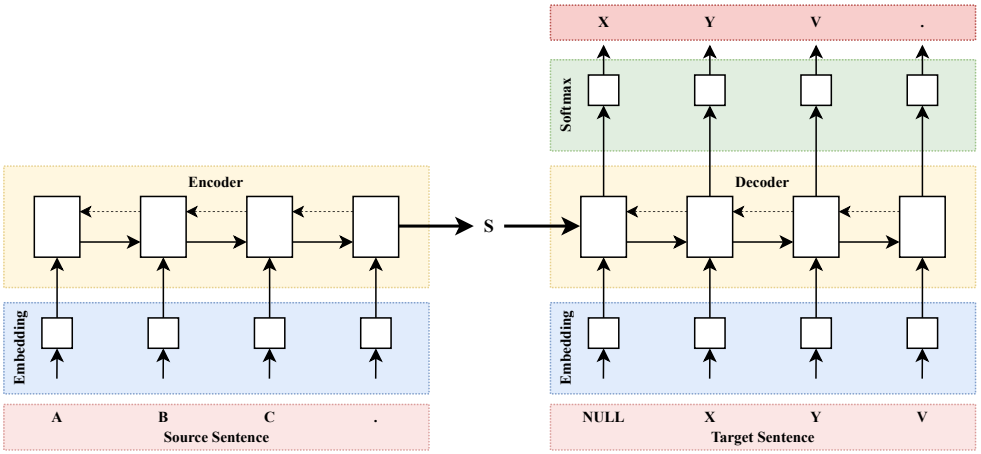


Figure 1. General structure of the Encoder-Decoder architecture.

Encoder–Decoder: The encoder network takes a source sentence and processes it word by word. The sentence passing through the encoder network is compressed into a fixed-length context vector [6]. The encoder network applies this to all hidden layers. The created context vector is given as an input to the decoder network. The decoder network reverses this function by translating it word by word into the target sentence. The encoder-decoder architecture creates a target sentence translation from the source sentence. It does not produce any additional output between the input and output sentences. This is also called end-to-end translation. When long sentences are compressed into a fixed-size vector, meaning is lost. One of the biggest problems of the first NMT systems is this loss of meaning. The attention mechanism is proposed to overcome this problem [1]. The attention mechanism provides additional position information and gives more accurate results in translating long sentences. Afterward, the greatest progress in NMT systems has been

with the introduction of the Transformer network [8]. When the results of the Machine Translation Conference (WMT20, WMT21) [2,9], are analyzed, it is seen that the models using the transformer architecture provide the best translation tasks. The system created in this study is based on the base transformer architecture. Transformer architecture will be explained in detail in the method section.

Pre-trained language models achieved the highest scores on most NLP tasks. One of the areas of research that has received the greatest attention is how to use language models most efficiently in natural language generation tasks, such as translation systems. Wu and Dredze [10] used the Zeroshot cross-lingual transfer method for tasks such as natural language inference, document classification, and named entity recognition. They conducted their study using the mBert model, which was trained in 104 languages. Karthikeyan et al. [11] used the M-BERT model for two different NLP tasks in the context of Spanish, Hindi, and Russian. In their studies, they explored the importance of lexical overlap between languages. Furthermore, Chi et al. [12] have concluded that multilingual models have a surprisingly high level of transfer and overlap. Guarasci et al. [13] explored the transferability of syntactic knowledge between languages, confirming the overlap in the aforementioned studies. Vries et al. [14] used the English-trained Bert model to investigate the effectiveness of transfer learning in low-resource languages and the implications of linguistic similarity in this process.

2.2. Turkish English Machine Translation

Machine translation from Turkish to English can be examined in 2 parts so far. Part 1 is Statistical Machine Translation systems and part 2 is Neural Machine Translation systems tested as a low source language pair. Part 1: In 2007 [15] and 2009 [16], carried out the first major studies for Turkish-English machine translation with SMT application. In these studies, they tried to solve sentence alignment and rare word problems by applying morphological word segmentation. Based on the Moses Statistical Machine Translation Toolkit [17], hierarchical expression-based models were presented in The International Conference on Spoken Language Translation (IWSLT) 2010 evaluation campaign. The SMT method is generated with an approach using a representation equivalent to the word form without splitting the Turkish words into morphemes. Generated the SMT method by using a representation equivalent to the word [18] form without splitting the Turkish words into morphemes and achieved 23.78 BLEU scores. The study prepared for the IWSLT 2013 conference was able to achieve 17.14 BLEU scores at the morpheme level and with a hierarchical phrase-based SMT system [19]. In 2019, a tree-based English Turkish statistical translation system was introduced [20]. The system prepared has achieved 21.4 BLEU scores. Part 2: In 2017, ref. [21] studied the IWSLT data set using the language model created with RNN in the deep fusion method with monolingual data. They received a score of 21.34 BLEU in the analysis using Neural Machine Translation. In the study [22] conducted to improve the translation quality due to the low resource available for Turkish-English machine translation, monolingual data were used with the training set, as in study [21], and reached 21.8 BLEU score with the IWSLT data set. Modern NMT systems were used in studies conducted in 2017, and solutions to the low-resource problem were pursued through methods such as transfer learning, byte pair encoding, and back-translation. In these studies, refs. [23–26] BLEU scores were obtained as 25.42, 18.9, 22.56, and 13.2 respectively. In the study in 2020 [27], a transformer-based Turkish-English NMT system was designed and a score of 28.48 BLEU was achieved. Created 5000-sentence dataset for Turkish Statistical machine translation [28] and a parallel corpus of 1M sentences were presented in the study prepared in 2021 [29].

3. Materials

The scarcity of parallel corpus can be considered one of the biggest challenges affecting the success of most machine translation systems. The parallel corpus can be defined as a corpus composed of sentence conjugates that have the same meaning in two different

languages. Besides, the content of the parallel corpus determines the type and quality of the translation system. The translation system that will be prepared with a collection of data from film subtitles or daily speeches may produce less accurate and academic terms translations. When the previous studies are reviewed, it is hardly possible to find a large-scale corpus for the Turkish-English or Turkish translation into another language. Academic articles and theses are one of the data sources used to create a parallel corpus [30]. That is why, to get an accurate and academic parallel data set, dissertations published in the "CoHE's Thesis System" in Turkey were used in this study.

3.1. Parallel Corpora Creation

In this study, we expand the work that we previously presented as a conference paper [29]. The abstracts of the dissertations are in English and Turkish languages. This study examined 245,100 dissertations published on CoHE's Thesis System site in different fields between 2013 and 2022. The theses are from a total of 186 different fields in CoHE's system. https://tez.yok.gov.tr/UlusalTezMerkezi/istatistikler.jsp (Accessed on 3 November 2022). Table 2 shows the details of data obtained from the theses on the first ten fields of study.

Table 2. The first 10 fields used to create the parallel corpus.

Fields	Percentage (%)
Education and Training	7.01
Business	6.04
Agriculture	2.85
Economy	2.63
Electrical and Electronics Engineering	2.37
Mechanical Engineering	2.35
History	2.33
Chemistry	2.2
Religion	2.19
Law	1.99

3.2. Sentence Alignment

Two different sentence alignment algorithms, Hunalign [31] and Vecalign [32] were used in this study. The results of these two algorithms were combined and overlapping sentences were selected.

3.2.1. Hunalign

Hunalign [31] aligns bilingual text at the sentence level. When there is a dictionary, Hunalign uses the bilingual text and combines this data with the data obtained from Gale-Church sentence length. When there is no dictionary, Hunalign first performs the sentence alignment considering the sentence length and then creates a dictionary based on this alignment. It then realigns the sentences using the created dictionary. In this study, the dictionary prepared by [33] was used.

3.2.2. Vecalign

Vecalign [32] attempts to align sentences using a new sentence alignment score function based on the similarity of bilingual sentence embeddings. The algorithm uses a pre-trained LASER multilingual (93 languages) sentence embedding [34] model for sentence embedding. Vecalign sentence alignment can be analyzed in two steps:

Step-1. The system calculates a score to indicate the probability that one or more adjacent source sentences and one or more adjacent target sentences are translations pairs,

Step-2. The alignment algorithm takes two documents and returns a hypothesis alignment using the score function calculated in the previous step.

It introduces a new scoring function based on the normalized cosine distance between multilingual sentence embeddings (Equation (1)).

$$cos(\Theta) = smilarity(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} \quad (1)$$

This function is implemented in conjunction with a new application of the dynamic programming approach [32] (Equation (2)).

$$c(x, y) = \frac{(1 - cos(x, y))nSents(x)nSents(y)}{\sum_{s=1}^{s}(1 - cos(x_s, y))} \quad (2)$$

in this equation:

x, y: source-target sentence,
$cos(x, y)$: similarity between embedding vectors,
$nSents(x), nSents(y)$: number of sentences in x and y,
$(x_1, \ldots, x_S), (y_1, \ldots, y_S)$: properly selected examples from the given document

3.2.3. Text Pre-Processing Steps

In this section, all preprocessing steps are explained in detail. Each dissertation published between the years 2013–2022 on the CoHE's site is handled one by one. In the details of each dissertation on the website, there are abstract sections in Turkish and English. These data are saved in a text file to create a corpus.

The text preprocessing steps are as follows:

1. The data were recorded to text files by classifying them according to their fields and publishing year on the CoHE's site.
2. A corpus was created by reading each file in turn. In this step, the same theses published with more than one field tag were filtered. Then the data is split into sentences. The data consists of approximately 3.5 M Turkish and English sentences at this stage.
3. The text has been translated into lowercase letters. Special characters and punctuation marks have been removed from the text so that it consists of numbers and letters only.
4. Sentences with dimensions greater than 400 characters and less than 20 characters were ignored.
5. The difference between the number of characters of Turkish and English sentences was filtered as a maximum of 150 characters.
6. The results of Hunalign and Vecalign sentence alignment algorithms were combined to provide separate sentence alignments for each thesis.
7. After all these processes, 1,217,300 sentences were obtained and approximately 2.3 M sentences were ignored.
8. 1000 sentences randomly selected from these sentences were checked by a supervisor. It was determined that the translations of only 2 sentences were not matching, and this was a tolerable ratio. While the matches were checked, the quality of the translation was ignored.
9. The created corpora consisted of approximately 30 k tokens for each language.

These sentences are randomly split for 10 k validations and 10 k tests (see Table 3).

Table 3. The Parallel Corpus.

Data	Sentences
Train	1,197,300
Validation	10,000
Test	10,000
Total	**1,217,300**

4. Methods

In this study, a transformer-based model is proposed for neural machine translation. To increase the quality of translations, methods such as shallow fusion, byte pair encoding, and beam search have been added to the created model (see Figure 2). In this section, the methods that make up the model and the evaluation criteria are explained in detail.

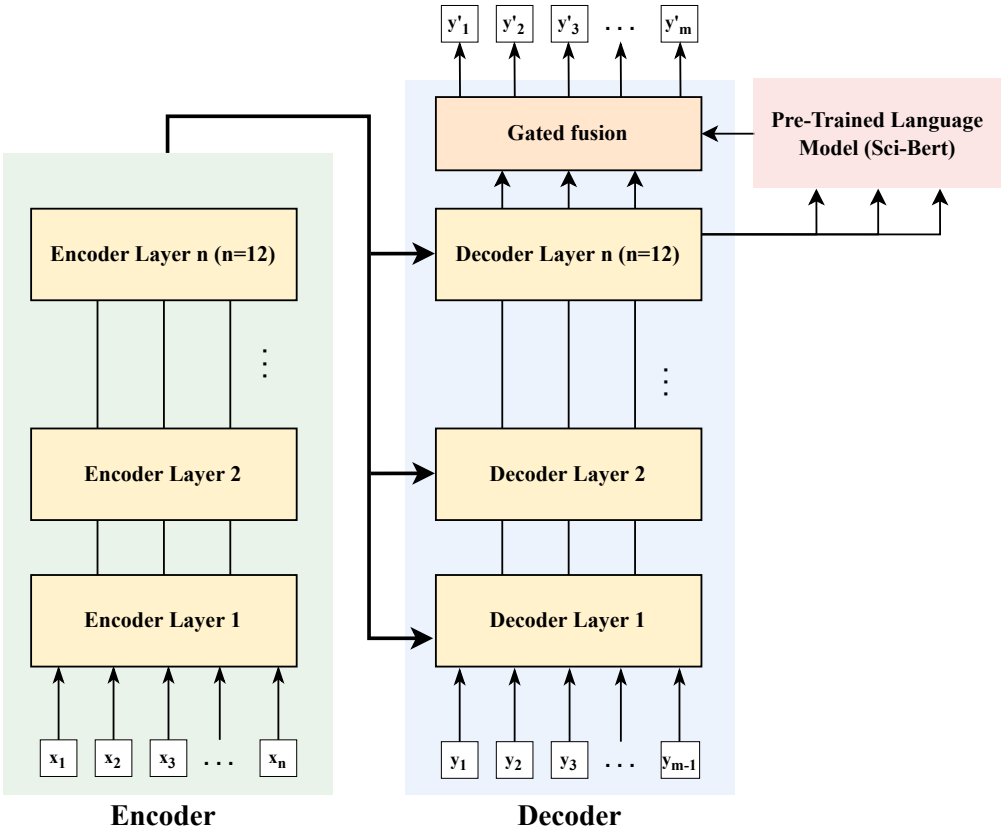


Figure 2. The Proposed Model.

4.1. Transformer Encoder-Decoder

The attention mechanism in machine translation tasks was first introduced by Bahdanau, Cho, and Bengio (2015) [1]. The attention mechanism forms a part of the neural architecture that enables the properties of the input data in text string form to be dynamically highlighted [35]. The attention mechanism became very popular in machine translations following this model released in 2015. In 2017, Vaswani et al. from the Google machine translation team [8] created the transformer model that has achieved the greatest success rate in machine translation so far by applying the attention mechanism in neural networks rather than RNN (see Figure 3).

The transformer model, like most NMT systems, is based on the Encoder-Decoder framework. The original article [8] used $N = 6$ and $N = 12$ layers for encoder and decoder.

Encoder: All encoder layers use the same structure and each layer is divided into two sub-layers: Self Attention (Multi-Head Attention) and Feed-Forward Neural Network (FFNN). The encoder input vector (Word embedding + positional coding) first passes through the self-attention layer, and this layer ensures that the relevant word appears along with other words in the sentence. The input vector is fed to the FFNN network with the output of this layer. Also, normalization is applied to all sublayer outputs. All sublayers and embedding layers produce $d_{model} = 512$ output.

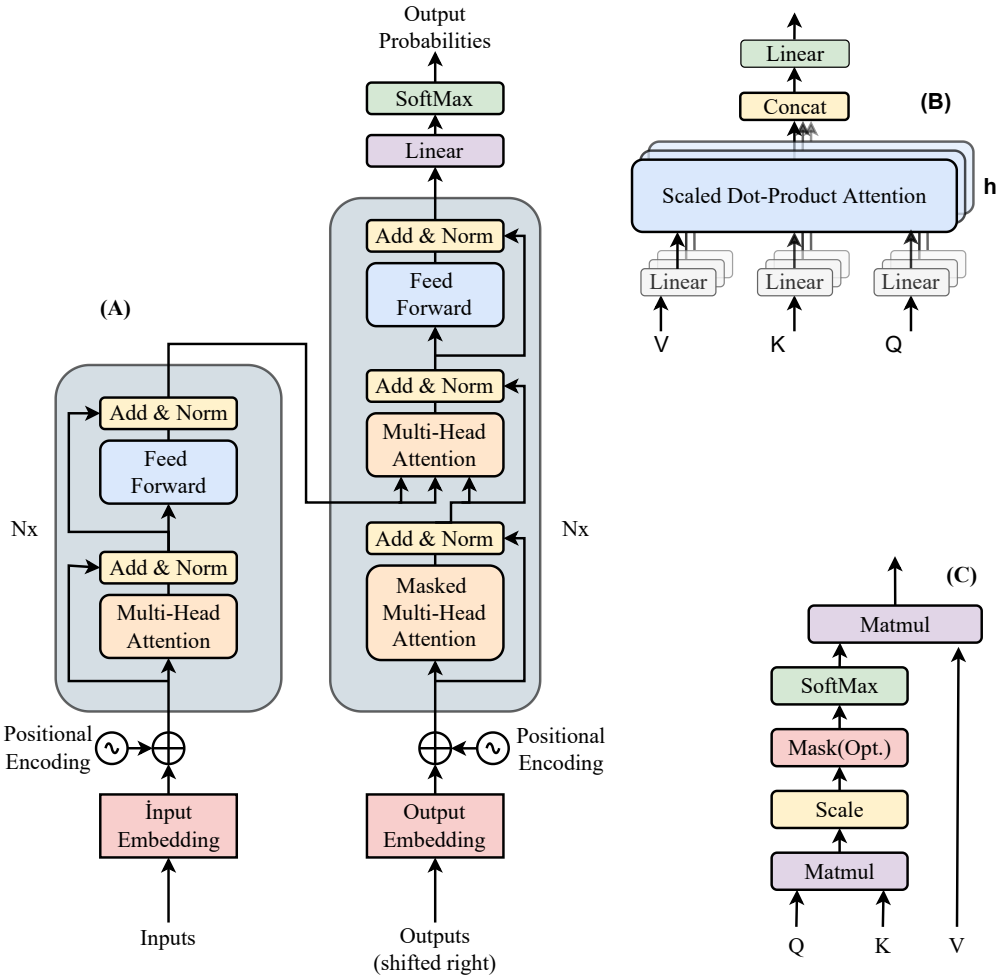


Figure 3. Transformer model architecture (**A**) Encoder-Decoder, (**B**) Multihead attention, (**C**) Scaled dot product.

Decoder: In addition to the two sublayers in the encoder layer, a multi-headed attention layer has been added. The output of the encoder network is given as input to each decoder network.

Self Attention: This layer aims to calculate the attention of all other words for a selected word. For this purpose, Query uses Key and Value vectors. The size of the vectors is 64. The Query is the word searched for when considering a database query. The value is the value corresponding to the searched word. The key is the result chosen from among the values. The attention score for each word is calculated as shown in (Equation (3)) and (Figure 3C). The softmax function is applied to obtain the weights on the attention output values.

$$Attention(Q, K, V) = softmax(\frac{QK^t}{\sqrt{d_k}})V \tag{3}$$

The attention score (head) for each word is more than 1. In the original article, $h = 8$ was determined. Independent attention outputs are combined and converted into a linear dimension (Equations (4) and (5) and Figure 3B). W is the trainable parameter vector.

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^0 \tag{4}$$

$$where\ head_i = Attention(QW_i^Q, QW_i^K, QW_i^V) \tag{5}$$

There is a feed-forward neural network after the multi-head attention layer in each encoder and decoder block (Equation (6)) and (Figure 3A).

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{6}$$

Unlike previous models, [1], the attention score is calculated for each word, instead of calculating just one attention score for the input sentence.

Positional Encoding: Since there is no repetition and convolution in the model, it is the process of adding some information about the relative or absolute position of the words in the sentence so that the model can understand the sentence order. Position encodings are the same size as Word embeddings and are easy to collect. *Pos* in (Equations (7) and (8)) shows the position and i dimension. Two calculated positions and word embeddings are summed up and transmitted to the encoder layer as input.

$$PE_{pos,2i} = sin(pos/10000^{2i/dmodel}) \tag{7}$$

$$PE_{pos,2i+1} = cos(pos/10000^{2i/dmodel}) \tag{8}$$

In the transformer model used in this study, encoder and decoder layers $N = 6$ and $N = 12$ were selected. In addition, FFNN the hidden layer is generated as 2048 and Word embedding d_{model} is chosen as 512. The vector size is set to 64 for the Query, Key, and Value.

4.2. Integration of the Pre-Trained Language Model into the Translation System

In this layer of the translation system, it is attempted to include grammar to make the target translation more fluent [21,36]. A pre-trained language model for English, the target language for this task, has been added to the system. Language models try to calculate the likelihood of the next word appearing during a word string [37]. Language models are trained with a massive monolingual text corpus. After the transformer architecture and especially the BERT [38] language model is used, successful results are obtained in most NLP tasks. Since we aimed to create an academic translation system, we used the SciBert [39] model created for NLP tasks in the scientific field in this study. This model included 1.14 M academic papers and consisted of 3.1B tokens in total. In translation systems, pre-trained language models are added using deep and shallow fusion methods. In the Deep Fusion Method, the language model is integrated into the decoder network. When calculating the output likelihood of the next word, the model is fine-tuned to use the hidden states of both the decoder network and the language model [21]. In the shallow fusion method, the output layer of the decoder network and the output layer of the language model are combined [40]. In this study, the shallow fusion method is used. A gate called the Shallow Gate is used to combine decoder network output with language model output. In the known shallow fusion method, [40] feed-forward network structure is suggested for this gate. In this study, FAN architecture is proposed in the design of the gate.

4.2.1. The Shallow Fusion with FFN

Pre-generated tokens are taken into account when calculating the probability of the occurrence of each token in translation systems. This is also the same for the decoder network and the language model (Equation (9)).

$$y' = \sigma_{trans}^{(t)} \cdot y'_{dec} \cdot \sigma_{lm}^{(t)} \cdot y'_{lm} \tag{9}$$

in this equation:

t_{th}: output token,
$y' = log\ p(y_t = k|x, y_{<t})$: final output,
$y'_{dec} = log\ p_{trans}(y_t = k|x, y_{<t})$: output of decoder network,
$y'_{lm} = log\ p_{trg}(y_t = k|x, y_{<t})$: output of language model network,

$\sigma_{trans}, \sigma_{lm} \in (0,1)$ given in the equation are the gates that allow the translation system to choose between the language model and the decoder network. The gates are a feed-forward network with one hidden layer and two outputs (Equation (10)).

$$[\sigma_{trans}^{(t)}, \sigma_{lm}^{(t)},]^{\top} = FFN(s_{trans}^{(t)}) \tag{10}$$

4.2.2. The Shallow Fusion with FAN

In the proposed method, the fully attentional network is used for the gated fusion layer. The Decoder network output and the pre-trained language model output are flat concatenated (Equation (11)).

$$y'' = concat([< cls >, y'_{dec}, < sep >, y'_{lm}]) \tag{11}$$

The generated sentence pairs (y'') are processed in token embedding, segment embedding, and position embedding layers. The sum of these processes is introduced as an input to the transformer encoder network (Equation (12)). (Figure 4).

$$y' = TransformerEncoder(E_{token}(y'') + E_{seg}(y'') + E_{pos}(y'')) \tag{12}$$

In this configuration, it is important to freeze language model weights during training and the translation model should be able to decide when and how much it can rely on the language model.

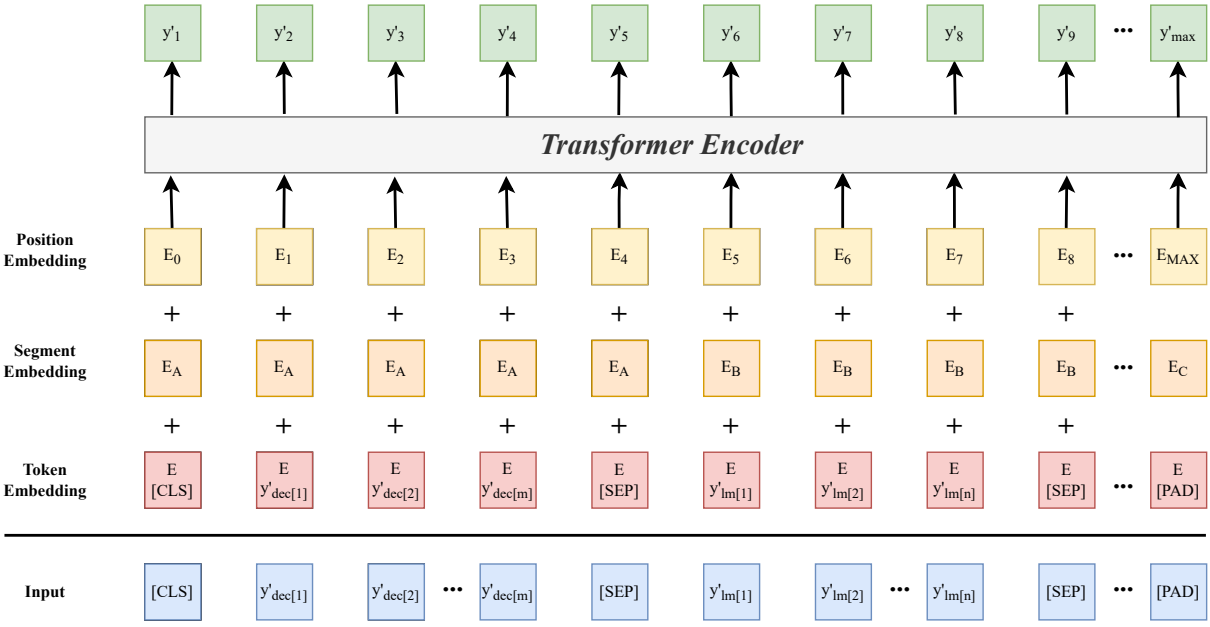


Figure 4. Gated Fusion (FAN).

4.3. Byte Pair Encoding (BPE)

Rare words in machine translation systems are one of the most important factors affecting translation quality [24]. To resolve this problem, Sennrich et al. [41] suggested representing the words by dividing them into subunits. The method that is inspired by a data compression algorithm is based on replacing the n characters that are most frequently used with another character that is not in the text. When the literature is analyzed, it is apparent that researchers often preferred this model in the latest machine translation systems to increase success. In this study, Byte Pair Encoding (BPE) was utilized to boost translation quality and solve rare words issue.

4.4. Beam Search

In NMT systems, search algorithms are not a required component, but they can have a big impact on the quality of the translation. Greedy Search and Beam Search are among the most commonly used algorithms. The latest studies have shown that Beam-Search outperformed the former algorithm [42]. Beam-Search stores the highest possible (k) amount of words to be generated. The next step is to combine each candidate word with the new word to create a potential new translation. This process continues until the translation is complete. The log probability for the most suitable selection of sentences is calculated. In this study, the Beam Search algorithm is used to calculate the final output.

4.5. Evaluation

There are many evaluation metrics in machine translation. In this study, Perplexity was used during the training. While testing the translations, metrics BLEU, METEOR, and TER were used.

Metrics like BLEU and METEOR are frequently used by researchers to assess the quality of translation. METEOR correlates better with human evaluation than BLEU [6]. However, it requires translation hypothesis tables with language-specific explanations. On the other hand, BLEU is easy to use and independent of language. Meteor and BLEU ratings were separately computed for this investigation.

4.5.1. Perplexity

In the translation system, the similarity between the target sentence probability distribution p and the candidate sentence probability distribution q is calculated with the help of cross entropy $(H(p,q))$. Perplexity is defined as the exponentiation of cross entropy [43] (Equation (13)).

$$Perplexity(p,q) = 2^{H(p,q)} \tag{13}$$

4.5.2. BLEU (Bilingual Evaluation Understudy)

BLEU score is an assessment criterion frequently used in machine translations. It aims to measure how similar machine translation is to human translation [44]. Typically, there are many "perfect" translations of a given source sentence. These translations may change in word selection or word order, even if they use the same words. This approach tries to match n-grams in the candidate text to n-grams in the reference text. Gram denotes words, so the comparison is made regardless of the word order. As the number of grams matches increases, the BLEU score gets higher at that rate (Equation (14)).

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')} \tag{14}$$

4.5.3. METEOR (Metric for Evaluation of Translation with Explicit Ordering)

It is an evaluation metric used to measure translation quality in machine translations [45]. METEOR first tries to align the unigrams between the candidate sentence and the reference sentence, and the longest alignment is chosen. Unigrams can be word-word, word-stem, or word-synonym. Each unigram in the candidate translation must match zero or one unigram in the reference sentence. *Precision* and *Recall* are calculated after unigrams are matched (Equations (15) and (16)).

$$P = \frac{m}{w_t} \tag{15}$$

$$R = \frac{m}{w_r} \tag{16}$$

In the equations, m is the number of unigrams in the candidate translation and matching the reference translation, w_t is the number of all unigrams in the candidate translation,

and w_r is the number of all unigrams in the reference translation. Precision and recall are combined with the harmonic mean (Equation (17)). Up to this stage, calculations evaluate matching unigrams. Nonadjacent matches in the candidate sentence and reference sentence should be added as a penalty (Equation (18)).

$$F_{mean} = \frac{10PR}{R + 9P} \tag{17}$$

$$p = 0.5(\frac{c}{v_m})^3 \tag{18}$$

In the equation, c is the number of parts v_m is the number of matching unigrams. Finally, the penalty score is applied to the harmonic mean (Equation (19)).

$$M = F_{mean}(1 - p) \tag{19}$$

4.5.4. TER (Translation Error Rate)

TER is calculated based on the amount of post-editing in the translation [46]. That is, it is the number of actions required to align the candidate sentence with the reference sentence. It is used regardless of language. The higher the score, the higher the error rate in the translation.

5. Result

5.1. Implementation Setup and Hyperparameters

The proposed model is shown in Figure 2. In addition to the base transformer method, a pre-trained language model was used. Byte Pair Encoding was used to solve the rare word problem. The Beam search algorithm was used after training the model to improve the translation quality. The hyperparameters of the generated models are given below (Table 4). Python programming language was used throughout the study. Selenium, NLTK, and SpaCy libraries were used in web scraping and text preprocessing steps. Pytorch deep learning framework was used for training the models. The pre-trained SciBert model was used through the Huggingface (https://huggingface.co/models) (Accessed on 3 November 2022) library. During the training of the models, the "adam" function was used as the activation function. In addition, the label "smoothing" was selected and the smoothing rate was determined as 0.1. Half precision (FP16) 16-bit floating-point was used to decrease the training time. The training was completed using GPU-based cloud computing systems and 4×Nvidia RTX 3090 graphics cards are used.

Table 4. Hyperparameters.

	Layer	**FFNN**	$\mathbf{d_{model}}$	**dQ, dK, dV**
Conv	20	1024	-	-
LSTM	2	1024	-	-
Transformer	6	2048	512	64
Transformer Big	12	2048	512	64
FFN	1	1024	-	-
FAN	4	2048	512	64

5.2. Evaluating Translation Quality

The parameter sizes of the created models are given (Table 5). A perplexity score was used to stop the training. If the perplexity score on the validation test set did not decrease for 10 epochs, the training was stopped. The final perplexity scores of the models in the validation and training datasets are given in Figure 5.

Table 5. Created model trainable parameter size.

	Parameter Size
Conv	96.841.408
LSTM	31.066.208
Transformer (6 layers)	73.033.074
Transformer Big (12 layers)	98.278.770
Transformer Big + FFN	136.303.045
Transformer Big + FAN	**158.523.484**

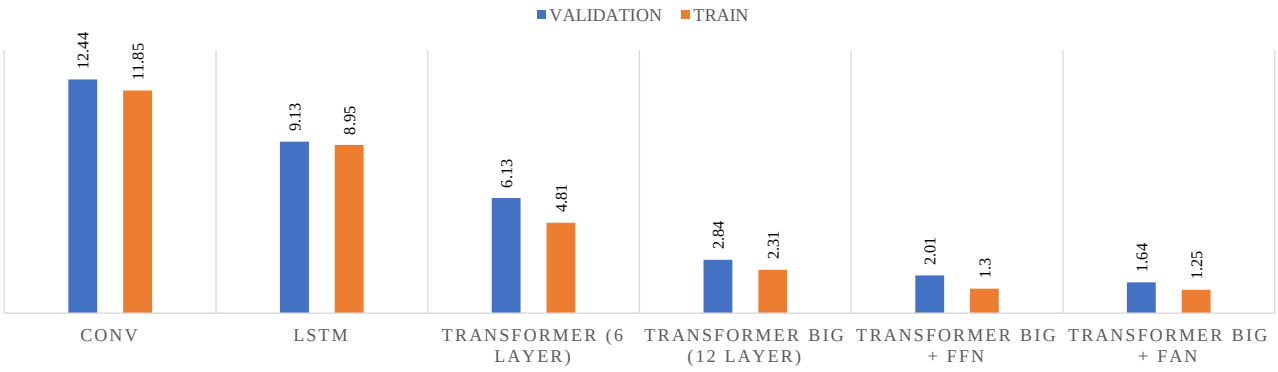


Figure 5. Perplexity score of the models created.

TER, BLEU, and METEOR scores of all models created were calculated separately. In addition, the test set was translated using google translate. Evaluation scores of this translation were also calculated. All results are shown in Figure 6.

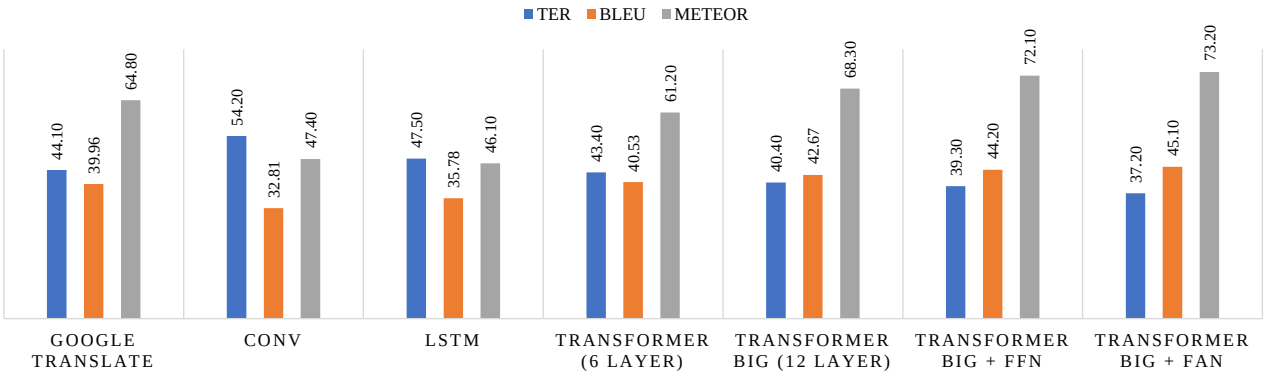


Figure 6. TER, BLEU, and METEOR scores of the models created.

The created parallel corpus, raw data, proposed model, and sample translations are available on GitHub page https://github.com/ilhamisel/tr-en_translate (Accessed on 3 November 2022). Furthermore, sample translations are available in the Appendix with examples from the WMT17 and WMT18 test sets.

6. Discussion and Future Works

In this study, pre-trained language models were used to improve the translation quality in low-source language pairs. Turkish-English language pair was chosen for the translation system. For the translation system, first of all, a parallel corpus of academic translations were created. While creating the parallel corpus, the summary sections of the theses prepared in Turkey were used. For the translation system, a new model consisting of an encoder-decoder network with transformer architecture, a pre-trained language model, and a FAN architecture is proposed. In the proposed model, the last layer of the basic transformer decoder network is combined with the last layer of the pre-trained language

model using FAN. In this study, unlike the known shallow fusion method, FAN architecture was used for the first time instead of FFN. In addition, Byte Pair Encoding was used to overcome the rare word problem during the training. After the model was trained, the best translation was tried to be selected among the alternative translations with the Beam Search algorithm. During the testing phase of the proposed model, TER, BLEU, and METEOR scores were calculated (Figure 5). The proposed model achieved (+2 BLEU, +4 METEOR) higher scores than the base transformer method. The BLEU score of 40 and above is known as high-quality translation https://cloud.google.com/translate/automl/docs/evaluate (Accessed on 3 November 2022). In addition, the proposed model also achieves higher scores than the Google Translate translation system. The main reason for this situation may be the academic sentence structure and the excess of academic terms in the test set. Using the proposed model, the translation was made using the zero-shot method on the WMT17 and WMT18 test sets. The model used current settings. In other words, extra parallel corpus, and monolingual data were not used in the proposed model while making these translations. The proposed model achieved a BLEU score of 20.12 and 20.56 on these test sets, respectively. In the latest studies, [47] achieved a 21.9 BLEU score for the WMT17 test set and [48] achieved a 23.1 BLEU score for the WMT18 test set. It is observed that the parallel corpus and monolingual data exclusive to the translation type were used in the mentioned studies. The proposed translation model reached competitive BLEU scores while the highest scores on these test sets were recorded in prior studies. Furthermore, instead of the SciBert model used in the translation systems, language models like BERT, DistilBert, and GPT were tried to translate WMT test sets. However, the results were much higher than expected. This may be due to the use of unlabelled data while training the pre-trained language models. It is estimated that WMT test sets were used while training the pre-trained language models. The model can be used as a helpful resource for researchers preparing articles in English. Although the model achieves sufficient scores, it has less success compared to translation systems that are widely used and continue to be developed (En-De, En-Fr, etc. . .). The main reason for these low scores is insufficient parallel corpus. Other language pairs currently have 50 M parallel corpus [9]. There is no corpus of this size among the Tr-En language pair yet. In future studies, it will be tried to create both a corpus and a translation system that can be used not only for academic translations but also for general translations.

With the proposed method, the pre-trained language model was used to increase the translation quality. This method produced better translations than the base transformer architecture. The use of pre-trained language models is common in obtaining state-of-the-art scores in natural language processing tasks. However, how to use pre-trained language models most effectively in natural language generation tasks is still an important research topic. Especially in low-resource translation systems, the effect of pre-trained language models needs to be increased. Also, future studies will focus on this problem.

Author Contributions: Conceptualization—İ.S.; Methodology, İ.S.; Resources—İ.S.; Software—İ.S. and D.H.; Supervision—D.H.; Writing—review and editing—D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Inonu University scientific research and coordination unit with the Project number FDK-2022-2925.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
2. Barrault, L.; Bojar, O.; Costa-Jussa, M.R.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Huck, M.; Koehn, P.; Malmasi, S.; et al. Findings of the 2019 conference on machine translation (wmt19). In Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), Florence, Italy, 1–2 August 2019; pp. 1–61.
3. Li, F.; Zhu, J.; Yan, H.; Zhang, Z. Grammatically Derived Factual Relation Augmented Neural Machine Translation. *Appl. Sci.* **2022**, *12*, 6518. [CrossRef]
4. Nakazawa, T.; Yaguchi, M.; Uchimoto, K.; Utiyama, M.; Sumita, E.; Kurohashi, S.; Isahara, H. Aspec: Asian scientific paper excerpt corpus. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 2204–2208.
5. Neves, M.; Yepes, A.J.; Névéol, A. The scielo corpus: A parallel corpus of scientific publications for biomedicine. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 2942–2948.
6. Stahlberg, F. Neural machine translation: A review. *J. Artif. Intell. Res.* **2020**, *69*, 343–418. [CrossRef]
7. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
9. Ranathunga, S.; Lee, E.S.A.; Skenduli, M.P.; Shekhar, R.; Alam, M.; Kaur, R. Neural machine translation for low-resource languages: A survey. *arXiv* **2021**, arXiv:2106.15115.
10. Wu, S.; Dredze, M. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 833–844. [CrossRef]
11. Wang, Z.; Mayhew, S.; Roth, D. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
12. Chi, E.A.; Hewitt, J.; Manning, C.D. Finding Universal Grammatical Relations in Multilingual BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5564–5577. [CrossRef]
13. Guarasci, R.; Silvestri, S.; De Pietro, G.; Fujita, H.; Esposito, M. BERT syntactic transfer: A computational experiment on Italian, French and English languages. *Comput. Speech Lang.* **2022**, *71*, 101261. [CrossRef]
14. de Vries, W.; Bartelds, M.; Nissim, M.; Wieling, M. Adapting Monolingual Models: Data can be Scarce when Language Similarity is High. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 1–6 August 2021; pp. 4901–4907. [CrossRef]
15. Oflazer, K.; Durgar El-Kahlout, İ. *Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007.
16. Bisazza, A.; Federico, M. Morphological pre-processing for Turkish to English statistical machine translation. In Proceedings of the 6th International Workshop on Spoken Language Translation: Papers, Tokyo, Japan, 1–2 December 2009.
17. Mermer, C.; Kaya, H.; Doğan, M.U. The TÜBİTAK-UEKAE statistical machine translation system for IWSLT 2010. In Proceedings of the 7th International Workshop on Spoken Language Translation: Evaluation Campaign, Paris, France, 2–3 December 2010.
18. Yeniterzi, R.; Oflazer, K. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 454–464.
19. Yılmaz, E.; El-Kahlout, I.D.; Aydın, B.; Özil, Z.S.; Mermer, C. TÜBİTAK Turkish-English submissions for IWSLT 2013. In Proceedings of the 10th International Workshop on Spoken Language Translation: Evaluation Campaign, Heidelberg, Germany, 5–6 December 2013.
20. Bakay, Ö.; Avar, B.; Yildiz, O.T. A tree-based approach for English-to-Turkish translation. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 437–452. [CrossRef]
21. Gulcehre, C.; Firat, O.; Xu, K.; Cho, K.; Bengio, Y. On integrating a language model into neural machine translation. *Comput. Speech Lang.* **2017**, *45*, 137–148. [CrossRef]
22. Sennrich, R.; Haddow, B.; Birch, A. Improving neural machine translation models with monolingual data. *arXiv* **2015**, arXiv:1511.06709.
23. Currey, A.; Miceli-Barone, A.V.; Heafield, K. Copied monolingual data improves low-resource neural machine translation. In Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2017; pp. 148–156.
24. Nguyen, T.Q.; Chiang, D. Transfer learning across low-resource, related languages for neural machine translation. *arXiv* **2017**, arXiv:1708.09803.
25. Firat, O.; Cho, K.; Sankaran, B.; Vural, F.T.Y.; Bengio, Y. Multi-way, multilingual neural machine translation. *Comput. Speech Lang.* **2017**, *45*, 236–252. [CrossRef]

26. Ataman, D.; Negri, M.; Turchi, M.; Federico, M. Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English. *arXiv* **2017**, arXiv:1707.09879.
27. Pan, Y.; Li, X.; Yang, Y.; Dong, R. Dual-Source Transformer Model for Neural Machine Translation with Linguistic Knowledge. *Preprints* **2020**, 2020020273. [CrossRef]
28. Yıldız, O.T.; Solak, E.; Görgün, O.; Ehsani, R. Constructing a Turkish-English parallel treebank. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 112–117.
29. İlhami, S.; Hüseyin, Ü.; HANBAY, D. Creating a Parallel Corpora for Turkish-English Academic Translations. *Comput. Sci.* **2021**, 335–340. [CrossRef]
30. Soares, F.; Yamashita, G.H.; Anzanello, M.J. A parallel corpus of theses and dissertations abstracts. In Proceedings of the International Conference on Computational Processing of the Portuguese Language, Canela, Brazil, 24–26 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 345–352.
31. Varga, D.; Halácsy, P.; Kornai, A.; Nagy, V.; Németh, L.; Trón, V. Parallel corpora for medium density languages. *Amst. Stud. Theory Hist. Linguist. Sci. Ser. 4* **2007**, *292*, 247.
32. Thompson, B.; Koehn, P. Vecalign: Improved sentence alignment in linear time and space. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 1342–1348.
33. Pavlick, E.; Post, M.; Irvine, A.; Kachaev, D.; Callison-Burch, C. The language demographics of amazon mechanical turk. *Trans. Assoc. Comput. Linguist.* **2014**, *2*, 79–92. [CrossRef]
34. Artetxe, M.; Schwenk, H. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 597–610. [CrossRef]
35. de Santana Correia, A.; Colombini, E.L. Attention, please! A survey of neural attention models in deep learning. *Artif. Intell. Rev.* **2022**, 1–88. [CrossRef]
36. Yan, R.; Li, J.; Su, X.; Wang, X.; Gao, G. Boosting the Transformer with the BERT Supervision in Low-Resource Machine Translation. *Appl. Sci.* **2022**, *12*, 7195. [CrossRef]
37. Mars, M. From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough. *Appl. Sci.* **2022**, *12*, 8805. [CrossRef]
38. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
39. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A pretrained language model for scientific text. *arXiv* **2019**, arXiv:1903.10676.
40. Skorokhodov, I.; Rykachevskiy, A.; Emelyanenko, D.; Slotin, S.; Ponkratov, A. Semi-supervised neural machine translation with language models. In Proceedings of the AMTA 2018 workshop on technologies for MT of low resource languages (LoResMT 2018), Boston, MA, USA, 21 March 2018; pp. 37–44.
41. Sennrich, R.; Haddow, B.; Birch, A. Neural machine translation of rare words with subword units. *arXiv* **2015**, arXiv:1508.07909.
42. Britz, D.; Goldie, A.; Luong, M.T.; Le, Q. Massive exploration of neural machine translation architectures. *arXiv* **2017**, arXiv:1703.03906.
43. Yin, X.; Gromann, D.; Rudolph, S. Neural machine translating from natural language to SPARQL. *Future Gener. Comput. Syst.* **2021**, *117*, 510–519. [CrossRef]
44. Dušek, O.; Novikova, J.; Rieser, V. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Comput. Speech Lang.* **2020**, *59*, 123–156. [CrossRef]
45. Lavie, A.; Agarwal, A. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In Proceedings of the Second Workshop on Statistical Machine Translation, Prague, Czech Republic, 23 June 2007; pp. 228–231.
46. Snover, M.; Dorr, B.; Schwartz, R.; Micciulla, L.; Makhoul, J. A study of translation edit rate with targeted human annotation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, Cambridge, MA, USA, 8–12 August 2006; pp. 223–231.
47. Behnke, M.; Heafield, K. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 2664–2674.
48. Pan, Y.; Li, X.; Yang, Y.; Dong, R. Morphological word segmentation on agglutinative languages for neural machine translation. *arXiv* **2020**, arXiv:2001.01589.