Multi-Task Modeling of Phonographic Languages: Translating Middle Egyptian Hieroglyphs

Philipp Wiesenbach[*], *Stefan Riezler*[†,*]

[*]Computational Linguistics & [†]IWR
Heidelberg University, Germany
{riezler,wiesenbach}@cl.uni-heidelberg.de

Abstract

Machine translation of ancient languages faces a low-resource problem, caused by the limited amount of available textual source data and their translations. We present a multi-task modeling approach to translating Middle Egyptian that is inspired by recent successful approaches to multi-task learning in end-to-end speech translation. We leverage the phonographic aspect of the hieroglyphic writing system, and show that similar to multi-task learning of speech recognition and translation, joint learning and sharing of structural information between hieroglyph transcriptions, translations, and POS tagging can improve direct translation of hieroglyphs by several BLEU points, using a minimal amount of manual transcriptions.

1. Introduction

The Middle Egyptian language was spoken for around 700 years, starting at around 2000 BCE, and is manifested on monuments (tombs, temples, stelas), ostracas (clay fragments often used by scribe apprentices) and papyri (mostly state administration documents, but also in the form of the well-known "book of the dead", letters and literature). Although the dry climate of the desert regions helped to preserve the sources, many cases of tomb pillaging, black market transactions, malicious destruction, and simply the ravages of time, reduced the possible amount of archaeological evidence. Thanks to efforts such as the Thesaurus Linguae Aegyptiae[1], a database that is administrated by the Berlin-Brandenburgische Akademie der Wissenschaften, digitized data of parallel textually encoded hieroglyphs (*hro*), transcriptions (*trans*), POS tags (*pos*) and German translations (*de*), are available for research. We use a dump of their database, that - after pre-processing and clearing out unusable data - contained $91,398$ parallel samples. At the time we conducted our experiment, hieroglyphic encodings (*hro*) were available for 30% of the database.[2] We therefore conducted our experiments with an extremely low-resource corpus of $29,296$ tuples.

Our goal is to build a neural machine translation (NMT) system that translates hieroglyphs directly, i.e., without requiring a separate transcription step, while using manual transcriptions and annotations only as means to improve the model during training time. While we are not aware of any prior work that addresses the severe data sparsity problem in the direct translation of Egyptian hieroglyphs, we take inspiration from a related, similarly under-resourced, problem of direct speech translation. The data sparsity issue in this problem has been successfully tackled by information sharing with larger related tasks using multi-task sequence-to-sequence learning techniques [1]. Similar to joint learning and sharing of structural information between the tasks of speech recognition and translation, we can share information between the tasks of hieroglyph transcription (i.e., the task of converting hieroglyphs into alphabetic symbols representing uniliteral hieroglyphs) and the task of direct translation of hieroglyphs (i.e., translation without the manual transcription step). Both tasks induce a segmentation and disambiguation of a sequence of hieroglyphs into source or target words. The usefulness of a manual transcription step is visible by a gap of 8 BLEU points between translating hieroglyphs directly and translation of manual transcriptions. Our experiments show that learning an automatic transcription model for a pipeline of transcription and translation suffers from the small size and the noise in the transcription data. However, integrating transcriptions (and related manual annotations by POS tags) into multi-task learning approaches yields improvements of several BLEU points for direct hieroglyph translation, showing that the structural signal inherent to these data can overcome the noise and be successfully used for improved direct hieroglyph translation.

2. The Middle Egyptian Language

2.1. Writing System

Egyptian written language was realized in two main forms. The first is hieroglyphic pictograms which are commonly known from wall paintings in Egyptian palaces and tombs. The second form is the hieratic script, a cursive version of the hieroglyphs, that was used to write letters, bills and administrative documents. Here, the text medium was mainly papyrus and ostraca. Our data cover both sources of hieroglyphs.

A hieroglyph can bear one of the following meanings:

---

[1] http://aaew.bbaw.de/tla/

[2] The reason that hieroglyphic encodings for the full dataset weren't available at the time we conducted our experiments is owed to the fact that the database creation is work in progress.

Gardiner Code	Sign(s)	Transcription	Description
G1	𓄿	ꜣ	Egyptian vulture
D36	𓂝	ꜥ	arm
D4	𓁹	jr	eye
D50	𓂭	ḏbꜥ	finger

Table 1: Example of Egyptian literals

1. Phonogram
2. Ideogram
3. Determinative

The phonographic meaning of hieroglyphs conveys 1-, 2-, 3- or (very rarely) 4-consonant sounds which can be combined following the rebus principle, i.e., the meaning is represented by the sound, instead of being abstracted from the single pictograms that the hieroglyph is constituted of.

Ideograms are signs that bear the visual meaning of the pictogram, paired with a mute vertical stroke. The image of a house □ without a stroke denotes the phonogram *pr* for usage in words like *prj* ("go out of smth.") or *pr-ꜥ ꜣ* ("palace", "pharao"). Paired with the vertical stroke it signifies an actual house (more specifically: the layout of the same).

The third function of a literal can be an unread determinative, denoting the semantic sphere of a word. The addition or the switch of a determinative can change the meaning or even the word class of a words. The following example (taken from [2]) displays two words that are pronounced as *mḥr*, with the last literal differentiating their meaning:

Figure 1: "pyramid"　　　Figure 2: "pain"

2.2. Transcription

The phonographic reading of hieroglyphs allows to interpret them as phonograms of consonants that can be transcribed into uniliteral signs of a conventionalized transcription language.[3] These signs do not specify the exact sounds of how hieroglyphs were pronounced, but are abstracted from how the Egyptian language has been conveyed in other languages (especially in Coptic). This phonemic alphabet, consisting of Latin and Hebrew letters, contains also "vowel-like" signs (ꜥ) and literals that denote *some* (unknown) vowel (*j*). Table 1 visualizes some of these signs together with their symbolic meanings. Column one denotes the Gardiner Code [3], a segmentation of the signs found in Middle Egyptian texts into 26 groups. Note

---

[3]We will not distinguish between a transliteration of graphemes and a transcription of phonemes and instead use the terms transliteration and transcription interchangeably.

Type	Source	Target
Nominal sentence	*sn.t=f ꜣs.t*	"His sister is Isis"
Adjectival sentence	*nfr sn.wt=f*	"His sisters are beautiful"
Adverbial sentence	*jnk m pr=f*	"I'm in his house"

Table 2: Nonverbal phrases

Type	Source	Target
Nominal subject	*jw sḏm zj*	"The man hears."
Pronominal subject	*jw sḏm=f*	"He hears."
Pronominal object	*jw sḏm sw zj*	"The man hears him."

Table 3: Verbal phrases

that the standard egyptological transcription does not capture determinatives: The transliterated representation will be the same for any word with different determinatives (as long there is no other reason to transcribe it differently).

2.3. Grammar

The classification between non-verbal and verbal phrases plays a major rule when deciphering Middle Egyptian. Non-verbal phrases can be distinguished into nominal, adjectival and adverbial phrases that do not contain an inflected verb. The predicative role to the subject noun is then taken by another noun, an adjectival or adverbial phrase. Example sentences are given in table 2, where "." denotes the separation of genus and number tokens.

As the name already reveals, verbal sentences introduce verbs that are inflected by suffixes (e.g. =f → "he"/"his"). Syntactic constituents can be classified by the basic rule of word-order, which is Verb-Subject-Object, although the order changes when objects become pronominal. Table 3 shows some examples. *jw*, in each case, initiates a main clause.

3. Data

3.1. Thesaurus Linguae Aegyptiae

Our work is based on a dump of the Thesaurus Linguae Aegyptiae (TLA) of the Berlin-Brandenburgische Akademie der Wissenschaften of 2018/01/30. The TLA project collects and edits Egyptian texts of different research groups in a database. The texts within the corpora are adapted to the Text Encoding Initiative[4] (TEI) and administrated within a schema-free database with many diverse attributes. All texts are tokenized and both hieroglyphic encoding and transcription are available. Each token possesses a link to a dictionary, where further information like POS tag and lemma is stored. The sentences are mostly translated into German, few into English or both. As stated in section 1, hieroglyphic encoding in Gardiner standard codes is available for around 30% of all sentences. The

---

[4]http://www.tei-c.org/index.xml

Gardiner encoding also includes special markers for spatial arrangement of the hieroglyphs as signs can be grouped in different ways. For example, the word for "heart" *jb* can be written ♡ or ♡ᴵ, depending on the available space and the writer's preferences. We deleted these markings as they were considerably predominant, so that the *hro* sources resulted in contiguous series of Gardiner signs.

The selection of sources contains papyri and inscription ranging from the Old to the New Kingdom, therefore covering the years from around 2500 to 1000 BCE. The text objects we used mainly date back to the Middle and Old Kingdom. Although the present paper aims to examine "classical" Middle Egyptian, grammar and vocabulary between former named epochs are similar. A small percentage of the texts may even contain Late Egyptian language.

3.2. Data Extraction

The data was extracted from an intermediate *json* file using *jq*[5]. Parsing the file resulted in $29,269$ parallel sentences, including hieroglyphs (*hro*), transcriptions (*trans*), POS tags for transcriptions (*pos*) and German translations (*de*). In addition to these data, $62,129$ tuples of *pos/trans/de* were available (where hieroglyphic encodings were absent). An example tuple is given in table 4.

Type	Alignment				
hro	D21 Y1	A1	D21 N35 A2	V31A	
trans	rḫ	=j	rn	=k	
pos	verb	pronoun	substantive	pronoun	
de	to know	I/my	name	your(s)	.

Table 4: Example paralled data

3.3. Textcritic Signs

The TLA project follows the TEI conventions for dealing with historical text objects. This especially affects textual witnesses from dead languages, as text objects could be (partially) destroyed, hardly readable or grammatically ambiguous. The textcritic signs used in the TLA corpus and their handling during pre-processing are illustrated in table 5.

Symbol	Meaning	Handling
()	defective	erase parenthesis and content
[]	lost	erase parenthesis
{}	surplus	erase parenthesis
<>	omitted	erase parenthesis
⌈⌉	damaged	erase parenthesis
⸢ ?	unclear	erase parenthesis

Table 5: Handling of textcritic markers

The overall aim was to keep as much information as possible. Only the information about defective passages had to be deleted as also comments and explanations of the translators were often mistakenly added in the same type of parentheses.

4. Multi-Task Learning

4.1. Multi-Task Setup

We followed the approach of [4] in our implementation of multi-task learning. During training, the system switches between multiple encoders/decoders, according to a probabilistic schedule that controls the expected ratio between main and assistance tasks. We denote the main task as hieroglyphic encoding to German (*hro2de*), whereas assistance tasks could be one of the following:

- transcription to German (*trans2de*)
- hieroglyphic encoding to transcription (*hro2trans*)
- hieroglyphic encoding to transcription POS tags (*hro2pos*)

As depicted in figure 3, models with one source and multiple targets are understood to be *one-2-many* systems, models with multiple sources and one target *many-2-one* systems, and models with multiple sources and multiple targets *many-2-many* systems. Our experiments in section 7 cover all of these variations. During runtime only one of the encoder/decoder pairs and their according inputs/outputs is active. The error backpropagation during learning is thus specific to the respective current task.
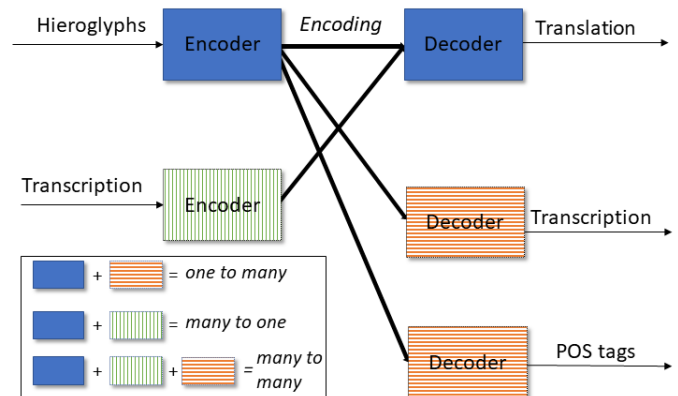
Figure 3: Possible multi-task learning setups

4.2. Learning Schedule

We adapt a multi-task schedule similar to [4] with a parameter α_i that denotes the average ratio between two tasks. When fixing $\alpha_1 = 1.0$ for the main task, the system switches to an assistance task$_i$ with probability $\frac{\alpha_i}{\sum_j \alpha_J}$. We employ $\alpha_i \in \{0.05, 0.10, 0.20, 0.30, 1.00\}$ to verify the amount of assistance data needed for the best results. When incorporating multiple assistance sources/targets, we didn't experiment with

different ratios for each tasks, but fixed α to be the same ratio for all of them.

4.3. Evaluation Technique

To reliably evaluate our learning setups, we applied 10-fold cross validation in every experiment series: Every fold was split into a disjoint train (90%) and test (10%) set, where we sampled another $1,000$ pairs from the train set as a held out validation set. After training our experiments for 200 epochs, we evaluated the system on the test set with parameters that achieved best results on the validation set. We report the average BLEU and the standard deviation from all ten runs.

5. Neural Architecture

Our implementation extended the machine translation framework Joey NMT [5], a neural toolkit written in Pytorch whose minimalist structure facilitated incorporating the multi-task learning schedule whereas still providing highly competitive performance. The base settings stayed the same for all experimental variations: We used an encoder/decoder sequence-to-sequence system with attention mechanism [6] that was fixedly attached to a decoder (and therefore was shared, when using many-2-one models). As encoders/decoders we employed GRU cells with a hidden size of 512 and a 20% dropout rate. For training we utilized ADAM as an optimizer with a learning rate of $2 \cdot 10^{-4}$. We trained every system for 200 epochs with a mini-batch size of 32 and tested from checkpoints that achieved best scores on a small holdout validation set. 10-fold cross validation helped to compensate bias of our low resource data. The only variable parameter was the embedding dimension for every data type. Here, we set 512 for *trans* and *de* respectively, 128 for *hro* and 32 for *pos*.

6. Baselines

Simultaneously to the multi-task experiment series, we trained several end-to-end systems, including a very strong transcription-to-translation upper bound. All runs besides back translation were repeated with 10-fold cross validation.

6.1. Hieroglyphs-to-Translation

The system that constitutes a baseline in our experiments is direct translation from hieroglyph script (*hro*) into to German text (*de*). The goal of our experiments is to improve the 19.77 BLEU (see table 6) by joint learning and information sharing with other tasks, however, avoiding a mandatory manual transcription step at test time.

	BLEU result	stddev
hro2de	19.77	1.11

Table 6: Results for *hro2de*.

6.2. Transcription-to-Translation

The system that translates manual transcriptions *trans* to German text *de* constitutes the upper bound in our experiments. Manual transcription incorporates both word boundaries and disambiguation of defects or variations in orthographic spellings. Table 7 shows that a gap of 8 BLEU points between *trans2de* and *hro2de* translation.

	BLEU result	stddev
trans2de	27.67	1.58

Table 7: Results for *trans2de*

6.3. Backtranslation

A common strategy to enhance learning in low resource sequence-to-sequence modeling scenarios is to augment the available corpus with synthetic samples [7]. Back-translation in our scenario meant to first train a *de2hro* system on the available parallel data, and create synthetic pairs to gradually enrich the basic *hro2de* sentence pairs. Table 8 shows BLEU results from 2,000 up to 10,000 additional samples. As can be seen, only backtranslation with the largest amount of data can slightly improve the results of the *hro2de* baseline.

Additional data	2,000	4,000	6,000	8,000	10,000
BLEU	19.42	19.67	19.39	19.37	20.57

Table 8: Results for various back-translation settings.

6.4. Pipeline

As pointed out in section 2.2, the transcription of hieroglyphs conveys phonograms, and as such also provides word boundaries. When considering hieroglyphs as a form of "speech", this transcription resembles the target output when training a speech recognition system. A straightforward baseline system could therefore build a pipeline to first translate the hieroglyphic encodings to the transcription (as in ASR) and then translate the generated transcription to the German target language (as in MT). We propose three training methods that all start from training an encoder/decoder model for *hro2trans* ($system_{base}$). Building on $system_{base}$,

1. train an encoder/decoder model for *trans2de* with the original training data for this pair ($system_1$);
2. train an encoder/decoder model for *trans2de* with the inferred training data from $system_1$ as source ($system_2$);
3. train an encoder/decoder model for *trans2de* with both the original from $system_{base}$ and the inferred training data from $system_1$ as source ($system_3$).

Results for all three setups are shown in table 9. As can be seen, none of the pipeline reaches the *hro2de* baseline results.

System	$system_1$	$system_2$	$system_3$
BLEU	18.97	19,86	19,76
stddev	1.40	1.80	1.76

Table 9: Pipeline results.

7. Multi-Task Experiments

In the following experiments, data for the main and the auxiliary tasks were taken from the 29, 269 tuples of *hro*, *trans*, *pos*, and *de* annotations.

7.1. Many-2-One

Our many-2-one multi-task experiment uses the transcription as additional input such that transcription and hieroglyphic encoding share the decoder for translation. Our reasoning why this setup could be beneficial is that during training of *trans2de*, the decoder is provided helpful information about word boundaries and syntax structure from the transcribed data, and keeps this as stored knowledge when switching back to *hro2de*. We experimented both with a basic (1 layer) and a deep (4 layers) and employed the ratios mentioned in section 4.2. Tables 10 and 11 show that up to $\alpha = 0.20$ nearly no improvements are recorded. Only with 4 layers we gain around 1 BLEU for $\alpha = 0.30$ and 2 BLEU for $\alpha = 1.00$.

α	0.05	0.10	0.20	0.30	1.00
BLEU	19.53	18.25	18.91	19.62	21.61
stddev	1.35	0.74	1.39	1.55	1.40

Table 10: Results for many-2-one with 1-layer architecture.

α	0.05	0.10	0.20	0.30	1.00
BLEU	20.38	19.82	20.06	20.41	21.60
stddev	1.25	0.63	1.28	1.55	1.24

Table 11: Results for many-2-one with 4-layer architecture.

7.2. One-2-Many

In our one-2-may setup, transcription and translation share the same encoder with hieroglyphic encodings as input. In these experiments we used the same settings as in section 7.1, but with the transcription as additional target language. The motivation for this setup is to condition the encoder on syntactic structure, word boundaries and disambiguation of varied pictogram compositions. The results shown in tables 12 and 13 show no improvements for 1-layer architectures and a gain of 2 BLEU for 4-layer structures.

To fully exploit all data available in these settings, we added an additional decoder for *hro2pos* pairs. We found that POS

α	0.05	0.10	0.20	0.30	1.00
BLEU	18.45	18.74	18.33	19.26	19.96
stddev	0.91	0.70	0.66	1.64	1.21

Table 12: Results for one-2-many with 1-layer architecture.

α	0.05	0.10	0.20	0.30	1.00
BLEU	19.99	20.31	20.03	20.78	21.92
stddev	1.46	2.06	1.18	1.18	1.45

Table 13: Results for one-2-many with 4-layer architecture.

α	0.05	0.10	0.20	0.30	1.00
BLEU	20.34	20.81	21.31	22.76	22.79
stddev	1.57	2.02	2.08	1.24	0.92

Table 14: Results for one-2-many with additional POS tags and 4-layer architecture.

information that allows to disambiguate between word classes offers complementary information to the structural information already provided by transcriptions. Table 14 highlights improvements of 3 BLEU in comparison to the *hro2de* baseline for $\alpha = 0.30$ and 4 layers. Remarkably, this improvement does not change when increasing α to 1.00, showing that it is sufficient to transcribe and tag 30% of the main data for optimal results.

In order to assess the contribution of *hro2pos* to the increase of BLEU found in table 14, we removed the *hro2trans* decoder and left *hro2pos* as sole assistance task. We found that this model showed a slight decrease 0.5 BLEU over the best result, demonstrating that both assistance tasks offer beneficial structural information. Additionally, we evaluated if the same results could have been achieved when solely using autoencoding, but this was not the case. Only 21.34 BLEU was reached for the assistance task *hro2hro*. Both results are listed in table 15.

Type	POS tags only	Auto-encoding
α	0.30	0.30
Layers	4	4
BLEU	22.38	21.34
stddev	1.64	1.58

Table 15: Results for one-2-many with POS tag only and auto-encoding

7.3. Many-2-Many

In this last experimental series, we allowed all connections as depicted in figure 3. We wanted to find out if the improvements gained from many-2-one and one-2-many settings could in some way accumulate to even better BLEU scores. As activating all connections caused relatively long run times, we only explored variations with $\alpha = 0.05$ and $\alpha = 0.30$. Results in table 16 for both of these settings showed that it was not possible to tune the model to benefit from the multiple tasks in a many-2-many setup. These results reflect the only marginal increases in BLEU from the corresponding many-2-many experiments reported in [4], where they achieved +0.5 BLEU when using autoencoding.

α	0.05	0.30
BLEU	17.78	18.07
stddev	1.31	1.46

Table 16: Results for many-2-many 4-layer architectures.

7.4. "All-in"

For this experimental series, we evaluated if our best *many2one* system could achieve even better results if it was provided all the available data pairings of *trans2de*. We therefore manipulated the algorithm to accept the same ratios as before, but created the assistance data iterator over all *trans2de* pairs. In this way, the amount of assistance data processed stayed the same, the data itself instead was taken from the complete set of parallel assistance pairs. The result was, that with a model of 4 layers, the maximum BLEU was reached earlier (at $\alpha_{all-in} = 0.30$ instead of $\alpha = 1.00$), but dropped again at $\alpha = 1.00$. This revealed that there was no improvement when the amount of assistance data was bigger than that of of the main task - no matter what ratio was employed. Table 17 summarizes these results.

α	0.30	1.00
BLEU	21.38	21.34
stddev	1.46	1.91

Table 17: 4-layer many-2-one system that iterates over all *trans2de* pairs within the scope of a certain switch ratio.

8. Analysis

In this section, we analyze translations from our best one-2-many system (mixing between *hro2de*, *hro2trans* and *hro2pos*) and compare them to translations of the baseline system *hro2de*. We found that indeed in many cases the improvements could be attributed to better segmentation of the hieroglyphic input sequence in the multi-task systems. Table 18 demonstrates the superior segmentation capabilities of the multi-task

system on a test set sample. Whereas *hro2de* interpreted D21 (r) incorrectly as the beginning of a new word r^c ("saying"), *one2many+pos* correctly split the input sequence between X1 (t in $k.t$ = "another") and F46 ($p\underline{h}r$ as triliteral in the beginning of $p\underline{h}r.t$) and produced the output "heilmittel", which is a valid translation for $p\underline{h}r.t$.

Another finding was that the training on multiple targets helped the system to correctly remember specific tokens. In the example of table 19 one can see that *hro2de* outputs the wrong name of the god mentioned in the source sentence. We conjecture that especially the transcription supports the system to memorize words in complex sequences.

hro	V31 X1 F46 D21 X1 N33 Z2
trans	kt p\underline{h}r.t
hde (reference)	ein anderes rezept .
hro2de	ein andere spruch .
one2many+pos	ein anderes heilmittel :

Table 18: Sample translation 1

hro	P6 D36 N35 G26B G7 D2 Ff100 Z1 . . .
trans	ꜥḥꜥ.n Ḏḥw.tj ḥr ḏd n Pꜣ-Rꜥ-Ḥr.w-ꜣḫ.tj.du
hde (reference)	da sagte thot zu reharachte :
hro2de	da sagte reharachte zu reharachte :
one2many+pos	da sagte thot zu reharachte :

Table 19: Sample translation 2

9. Conclusion

We presented an approach to direct translation of Middle Egyptian hieroglyphs that circumvents the need for segmentation and disambiguation via manual transcription at test time. Instead, we show that adding manual transcriptions and POS tags in multi-task training at an amount of 30% of the parallel hieroglyph data is sufficient to boost translation performance by 3 BLEU points, amounting to a 40% error reduction relative the upper bound of translation from manual transcriptions. This approach outperforms by far a straightforward pipeline that attempts to automatically transcribe hieroglyphs before translation. Our approach thus shows that sharing of structural information between related tasks is beneficial even in tasks that are too under-resourced to allow to build straightforward processing pipelines.

10. References

[1] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly transcribe foreign speech," in *Proceedings of Interspeech*, Stockholm, Sweden, 2017.

[2] C. Maderna-Sieben, *Mittelägyptische Grammatik für An-*

fänger - Ein ausführliches Kompendium für den Unterricht. Münster: LIT Verlag Münster, 2016.

[3] A. Gardiner, *Egyptian Grammar: Being an Introduction to the Study of Hieroglyphs*, 3rd ed. Oxford: Griffith Institute, 1957.

[4] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

[5] J. Kreutzer, J. Bastings, and S. Riezler, "Joey nmt: A minimalist nmt toolkit for novices," in *EMNLP-ICJNLP 2019: System Demonstrations*, Hong Kong, 2019.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

[7] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, Berlin, Germany, 2016.