




Neural Machine Translation for Amharic-English Translation

Andargachew Mekonnen Gezmu¹^a, Andreas Nürnberger¹^b and Tesfaye Bayu Bati²^c

¹Faculty of Computer Science, Otto von Guericke Universität Magdeburg, Universitätsplatz 2, Magdeburg, Germany

²Faculty of Informatics, Hawassa University, Hawassa, Ethiopia

Keywords: Neural Machine Translation, Low-resource Language, Subword.

Abstract: This paper describes neural machine translation between orthographically and morphologically divergent languages. Amharic has a rich morphology; it uses the syllabic Ethiopic script. We used a new transliteration technique for Amharic to facilitate vocabulary sharing. To tackle the highly inflectional morphology and to make an open vocabulary translation, we used subwords. Furthermore, the research was conducted on low-data conditions. We used the transformer-based neural machine translation architecture by tuning the hyperparameters for low-data conditions. In the automatic evaluation of the strong baseline, word-based, and subword-based models trained on a public benchmark dataset, the best subword-based models outperform the baseline models by approximately six up to seven BLEU.

1 INTRODUCTION

High-quality literary machine translation is not yet achieved with the current neural machine translation (NMT) models. Nonetheless, the models are quite useful and proved to be productive at least for tasks for which a rough translation is adequate. Good examples are information access on the Web and computer-aided human translation.


Despite its usefulness, NMT requires large training data to build competitive models. It may outperform phrase-based statistical machine translation (PBSMT) if the training data are more than 500 thousand parallel sentence pairs (Koehn and Knowles, 2017; Lample et al., 2018). However, if the system architecture is adapted to low-resource settings and hyperparameters are tuned for low-data conditions, then NMT may outperform PBSMT with as few as five thousand sentence pairs (Sennrich and Zhang, 2019).


In machine translation, we should also pay attention to the morphology of languages. Amharic, a Semitic language and lingua franca in Ethiopia, is one of the languages that have highly inflectional morphology. In Amharic, an orthographic word may represent a phrase, clause, or sentence. For example,


the word አስከፊብረረላቸው /iskiyabraralacəw/ meaning “until he explains it to them” is a clause. This word does not appear even once in the 24 million tokens Contemporary Amharic Corpus¹ (Gezmu et al., 2018). Yet its constituent stem and affixes – አስ/isk/-ይ/y/-አብረረ/abrara/-ል/l/-አቸው/acəw/ – appear several times in the corpus being part of other words. Here the lexical word is አብረረ /abrara/; the other grammatical words are attached to the lexical word. The alignment of Amharic words to English often appears to be one-to-many. Thus, the vocabulary of the language is too large for NMT. For open vocabulary NMT, using subword units has a vital role.

Moreover, as seen in the above example, the boundaries of morphemes are more easily identified in the transliteration than in the original script. Any word segmentation technique can take advantage of the Amharic transliteration. The transliteration also helps for sharing vocabulary with English especially loan words and named entities.

Therefore, in this research, we deal with a case of NMT between distant languages, Amharic and English, with regard to their orthography and morphology. It also deals with NMT in low-resource conditions.

^a <https://orcid.org/0000-0002-3424-755X>

^b <https://orcid.org/0000-0003-4311-0624>

^c <https://orcid.org/0000-0001-9042-7996>

¹ Available at: <http://dx.doi.org/10.24352/ub.ovgu-2018-144>

2 TRANSLITERATION

Amharic uses the Ethiopic writing system. Even though Ethiopic shares some features of abugida, the Unicode Consortium (2020) considers it as “simple syllabary”. The ancient Ethiopian Semitic language, Ge’ez, originally uses the writing system. Ge’ez is now extinct and used only for Liturgy.

Each character of the Ethiopic writing system is formed by systematic integration of a consonant and vowel (e.g., መ /mə/ and ሙ /mu/). Sometimes consonants and vowels can be written as bare consonants (e.g., ሙ /m/ and ን /n/) or bare vowels (e.g., አ /a/ in አለም /aləm/). In addition to the characters in the basic script set, some characters represent labialized variants of consonants followed by particular vowels.

There are also some homophonic characters in the writing system (e.g., ስ and ሙ represent /sə/ sound). Originally, these characters had distinct sounds but were lost in Amharic (Aklilu, 2004); they are the source of many cognates. For instance, ስጦ and ሙጦ are transliterated as /səm/ meaning “wax”. In modern use, the homophonic characters are used interchangeably. For consistent spelling, the Ethiopian Languages Academy proposed a spelling reform (Aklilu, 2004). According to the reform, homophonic characters, being redundant, should be reduced to their standard forms. For example, instead of ሙ /sə/ the character ስ /sə/ should be used. Also, some labiovelars are substituted by their closest counterparts in the basic script set (e.g., ቆ by ቁ /qu/).

There is no case difference in Amharic. Unlike other Semitic languages, such as Arabic and Hebrew, it is written from left to right and its orthography is nearly phonetic (Cowley, 1967).

There is no standard transliteration for Amharic. In line with its unique features and the reform of the writing system, we used a new transliteration scheme, Amharic transliteration for machine translation (AT4MT). To make it useful for machine translation, our objective was to transliterate Amharic loan words and named entities as close as the spelling of English words. In doing so, we had to consider the restoration of the original spelling to make it invertible. The algorithm follows a rule-based approach². It is similar to the one shown in Algorithm 1 for converting an Ethiopic numeral into an Arabic numeral. It maps Ethiopic characters to their phonemic representations in Latin-based characters. Table 1 demonstrates the transliterations of some Amharic words.

Table 1: Sample transliterations of Amharic words.

Amharic	AT4MT	English
ሆስፒታል	hospital	hospital
አንገላ	angela	Angela
ማስክ	mask	mask
ኢራን	iran	Iran
ኢራቅ	iraq	Iraq
አስራኤል	israel	Israel
ራዲዮ	radiyo	radio

Though in the modern-day Amharic writings Ethiopic numerals tend to be replaced by Arabic numerals, still they are in use. Like Roman numerals, the Ethiopic number system does not use zero or digital-positional notation. A number is represented by a sequence of powers of 100, each preceded by a coefficient equivalent to 2 through 99. For example, the number 345 is represented by $3 \cdot 100^1 + (40 + 5) \cdot 100^0 = 3 \ 100 \ 40 \ 5 = \overline{፫፻፵፭}$. Algorithm 1 converts an Ethiopic numeral into an Arabic numeral.

Algorithm 1: An algorithm that converts an Ethiopic numeral into an Arabic numeral.

```

ethiopic_arabic_table = { ፩:1, ፪:2, ፫:3, ፬:4, ፭:5, ፮:6,
፯:7, ፰:8, ፱:9, ፲:10, ፳:20, ፴:30, ፵:40, ፶:50,
፷:60, ፸:70, ፹:80, ፺:90, ፻:100, ፷፻:10000 }

FUNCTION ethiopicnum2arabicnum(number):
  IF LENGTH(number) == 1:
    RETURN ethiopic_arabic_table[number]
  result = 0
  FOR digit IN number:
    IF (digit == ፫) OR (digit == ፷፻):
      result = result * 100
    ELSE:
      result = result + ethiopic_arabic_table[digit]
  RETURN result

```

Transliteration of Amharic punctuation is a straightforward process. Word boundary is traditionally indicated by a colon like character (“:”); in modern use, a white word space is becoming common. The end of sentence marker is a double-colon like character (“::”) and is transliterated as a period “.”. A comma, semicolon, colon, hyphen, and question mark are “;”, “;”, “;”, “-”, and “?” or “?”, respectively; they have transliterated accordingly.

² The implementation is available at: <https://github.com/andmek/AT4MT>

3 NMT SYSTEM

While Recurrent Neural Network (RNN) based architectures (Sutskever et al., 2014; Bahdanau et al., 2014) have been used for NMT to obtain good results, the transformer-based ones are even enjoying better success (Vaswani et al., 2017). The transformer-based NMT system uses stacked layers to compute representations of its input and output without using RNNs. It uses a stack of identical, self-attention, and fully connected layers for both the encoder and decoder. Each layer has sub-layers. The first sub-layer is a multi-head self-attention mechanism and the second is a simple feed-forward network. In addition to the two sub-layers, as in each encoder layer, the decoder uses a third sub-layer, which performs multi-head attention over the output of the encoder stack.

Because of the long training times of NMT models, we followed best practices of prior research in low-resource settings instead of working with all possible architectures and hyperparameters. In RNN based NMT systems, there are mixed opinions on the size of training batch sizes in low-data conditions. While Morishita et al. (2017) and Neishi et al. (2017) are using large batch sizes, Sennrich and Zhang (2019) recommend small batch sizes. There is also a trend to use smaller and fewer layers (Nguyen and Chiang, 2018).

Table 2: Differences between transformer-tiny, transformer-small, and transformer-large; where TT is transformer-tiny, TS is transformer-small, and TL is transformer-large.

Hyperparameter	TT	TS	TL
Batch size	1024	4096	1024
No. of heads	4	4	8
No. of hidden layers	2	2	6
Filter size	512	512	2048
Hidden size	128	128	512
Learning rate constant	2.0	2.0	0.1
Warmup steps	8000	8000	16000

Therefore, we used three different hyperparameter set: transformer-large, transformer-small, and transformer-tiny. All systems use Adam optimizer (Kingma and Ba, 2014), dropout (Srivastava et al., 2014) rate of 0.1, and label smoothing (Szegedy et al., 2016) of value 0.1. Table 2 details the differences between the three systems. Transformer-small and transformer-tiny differ only in training batch sizes. Training batch sizes are given in terms of source and target language tokens.

4 PBSMT BASELINE

Our PBSMT baseline system had settings that were typically used by Ding et al. (2016), Williams et al. (2016), and Sennrich and Zhang (2019). We used the Moses (Koehn et al., 2007) toolkit to train PBSMT models. For word alignment, we used GIZA++ (Och and Ney, 2003) and the grow-diag-final-and heuristic for symmetrization. We used the phrase-based reordering model (Koehn et al., 2003) with three different orientations: monotone, swap, and discontinuous in both backward and forward directions, being conditioned on both the source and target languages. We removed sentence pairs with extreme length ratios and sentences longer than eighty tokens.

We used five-gram language models smoothed with the modified Kneser-Ney (Kneser and Ney, 1995). KenLM (Heafield, 2011) language modeling toolkit was engaged for this purpose. We have not used extra big monolingual corpora for language models. They are no longer the exclusive advantages of PBSMT as NMT can also be benefited from them (Sennrich and Zhang, 2019).

The feature weights were tuned using MERT (Och, 2003). We also used k-best batch MIRA for tuning (Cherry and Foster, 2012) by selecting the highest-scoring development run with a return-best-dev setting.

In decoding, we applied cube pruning (Huang and Chiang, 2007), Minimum Bayes Risk decoding (Kumar and Byrne, 2004), a distortion limit of six, and the monotone-at-punctuation (do not reorder over punctuation) heuristic (Koehn and Haddow, 2009).

5 EXPERIMENTS

We carried out the experiments in three scenarios. In the first scenario, we evaluated the performance of our three systems: transformer-tiny, transformer-small, and transformer-large. Then we evaluated our transliteration technique in the second scenario. Finally, we made a comparison of word-based and subword-based models in the third scenario. For the last two scenarios, we employed the best performing system in the first scenario. The same datasets were used in all scenarios; preprocessing, training, and evaluation steps were also similar.

5.1 Datasets

We trained our models on the public benchmark dataset, Amharic-English parallel corpus³, consisting of 140 thousand sentence pairs. The development and test sets have 2864 and 2500 sentence pairs.

5.2 Preprocessing

All the Amharic datasets were transliterated with AT4MT; they were tokenized with an in-house tokenizer. The English data sets were tokenized with Moses' script (Koehn et al., 2007).

For word-based models, we used a shared vocabulary of the top 44000 most frequent tokens. For subword-based models, we used word-piece (Wu et al., 2016), byte pair encoding (BPE) (Sennrich et al., 2016), and unigram language model (ULM) (Kudo, 2018). For all techniques, the segmentation models were built using the training datasets of both languages separately. We used the word-piece implementation in Google's Tensor2Tensor library (Vaswani et al., 2018); we used the BPE and ULM implementations in Google's sentence-piece library (Kudo and Richardson, 2018). In this implementation, we provide the desired vocabulary size for BPE instead of the number of merge operations as in its original implementation. To make a comparison among the segmentation schemes, tokens were segmented into 32000 subword vocabularies using word-piece, BPE, or ULM.

5.3 Training and Decoding

We trained each NMT model for 250 thousand steps. For decoding (inference), we used a single model obtained by averaging the last twelve checkpoints. Following Wu et al. (2016), we used a beam search with a beam size of four and a length penalty of 0.6.

5.4 Evaluation

Eventually, translation outputs of the test sets were detokenized and evaluated with a case-sensitive BLEU metric (Papineni et al., 2002). For consistency, we used Post's (2018) implementation of the metric, sacreBLEU⁴. To compensate for the limitations of BLEU (Callison-Burch et al., 2006; Reiter, 2018), we also used BEER (Stanojevic and Sima'an, 2014) and CharacTER (Wang et al., 2016) metrics. The

Amharic outputs were not back transliterated to use these automatic metrics effectively.

6 RESULTS

Tables 3 and 4 show the performance results of the three systems with BLEU, BEER, and CharacTER metrics in English-to-Amharic and Amharic-to-English translations. In this scenario, only the word-piece technique was used for word segmentation. The transformer-tiny model is the least performing model. The only difference between transformer-small and transformer-tiny is their training batch sizes. By just increasing the training batch size from 1024 to 4096, we gained more than one BLEU score for both English-to-Amharic and Amharic-to-English translations. BEER and CharacTER scores also reflect similar improvements. Unlike BLEU and BEER, the smaller the CharacTER score, the better. Transformer-large outperforms both systems.

Table 3: Performance results of transformer-tiny, transformer-small, and transformer-large in English-to-Amharic translation.

NMT System	BLEU	BEER	CharacTER
Transformer-tiny	17.8	0.485	0.639
Transformer-small	18.9	0.498	0.614
Transformer-large	26.7	0.552	0.523

Table 4: Performance results of transformer-tiny, transformer-small, and transformer-large in Amharic-to-English translation.

NMT System	BLEU	BEER	CharacTER
Transformer-tiny	24.0	0.523	0.629
Transformer-small	25.4	0.530	0.614
Transformer-large	32.2	0.570	0.539

In this case of low-data condition, using smaller and fewer layers are not beneficial for transformer-based systems; smaller batch sizes are not useful either.

Since the transformer-large is the best performing system, we have used it to evaluate word-based and subword-based models. Tables 5 and 6 show the performance results of the word-based and subword-based models. All subword-based models outperform word-based ones by approximately three up to four BLEU. In table 5, in the English-to-Amharic translation, the word-piece technique has the best performance; BPE and ULM have almost equivalent

³ Available at: <http://dx.doi.org/10.24352/ub.ovgu-2018-145>

⁴ Signature BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.9

performances. Besides, a better result was achieved when transliteration is applied to Amharic data. In table 6, in Amharic-to-English translation, the ULM technique has the best performance; word-piece and BPE have comparable performances.

Table 5: Performance results of word-based NMT, subword-based NMT, and PBSMT baseline models in English-to-Amharic translation.

NMT Model	BLEU	BEER	CharacTER
Word-based	23.0	0.510	0.592
BPE	25.7	0.547	0.525
ULM	25.6	0.548	0.524
Word-piece	26.7	0.552	0.523
Word-piece-no-transliteration	26.0	0.546	0.528
PBSMT-MERT	20.2	0.502	0.646
PBSMT-MIRA	19.4	0.485	0.702

Table 6: Performance results of word-based NMT, subword-based NMT, and PBSMT baseline models in Amharic-to-English translation.

NMT Model	BLEU	BEER	CharacTER
Word-based	29.1	0.537	0.592
BPE	32.7	0.571	0.534
ULM	33.2	0.578	0.527
Word-piece	32.2	0.570	0.539
PBSMT-MERT	25.8	0.508	0.633
PBSMT-MIRA	23.3	0.497	0.701

For the baseline PBSMT models, better scores were achieved when feature weights were tuned using MERT than MIRA. Thus, we took PBSMT-MERT as our strong baseline. The best subword-based models outperform the baseline models by approximately six up to seven BLEU.

7 CONCLUSIONS

We conducted Amharic-English NMT with complex morphology in low-resource conditions. Amharic has a rich morphology and uses the Ethiopic script. We used a new transliteration method that is useful for machine translation. To tackle the complex morphology and to make an open vocabulary translation, we used subwords. To segment words into subwords, we used word-piece, BPE, and ULM. Furthermore, based on the best practices of prior research in this line of work, we conducted NMT in low-data conditions. Thus, we used the transformer-based NMT architecture by tuning the hyper-parameters for low-data conditions. In this case of low-data condition, using smaller and fewer layers

hurts the performance of the transformer-based system; smaller batch sizes are not beneficial either. In the evaluation of word-based and subword-based models trained on a benchmark dataset, all subword-based models outperform word-based ones by approximately three up to four BLEU. Moreover, the best subword-based models outperform the baseline models by approximately six up to seven BLEU.

We urge on using the universal transformer-based architecture (Dehghani et al., 2019), auxiliary data like monolingual corpora, and other word segmentation techniques for further research. We also recommend a study on the syntactic divergence of the languages.

ACKNOWLEDGEMENTS

We would like to thank Nirayo Hailu and Tirufat Tesfaye for their support.

REFERENCES

- Aklilu, A. (2004). Sabean and Ge'ez symbols as a guideline for Amharic spelling reform. In *Proceedings of the first international symposium on Ethiopian philology*, pages 18–26.
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *Computing Research Repository*, arXiv:1409.0473. version 7.
- Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.
- Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montreal, Canada. Association for Computational Linguistics.
- Cowley, R. (1967). The standardization of Amharic spelling. *Journal of Ethiopian Studies*, 5(1):1–8.
- Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, Ł. (2019). Universal transformers. In *International Conference on Learning Representations*, pages 181–184, IEEE.
- Ding, S., Duh, K., Khayrallah, H., Koehn, P., and Post, M. (2016). The JHU machine translation systems for WMT 2016. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 272–280, Berlin, Germany. Association for Computational Linguistics.

- Gezmu, A. M., Seyoum, B. E., Gasser, M., and Nürnberger, A. (2018). Contemporary Amharic corpus: Automatically morpho-syntactically tagged Amharic corpus. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 65–70, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.
- Huang, L. and Chiang, D. (2007). Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic. Association for Computational Linguistics.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *Computing Research Repository*, arXiv:1412.6980. version 9.
- Kneser, R., and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, Volume 1, pp. 181–184. IEEE.
- Koehn, P. and Haddow, B. (2009). Edinburgh’s submission to all tracks of the WMT 2009 shared task with reordering and speed improvements to Moses. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 160–164, Athens, Greece. Association for Computational Linguistics.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.
- Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.
- Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.
- Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Kumar, S. and Byrne, W. (2004). Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.
- Morishita, M., Oda, Y., Neubig, G., Yoshino, K., Sudoh, K., and Nakamura, S. (2017). An empirical study of minibatch creation strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 61–68, Vancouver. Association for Computational Linguistics.
- Neishi, M., Sakuma, J., Tohda, S., Ishiwatari, S., Yoshinaga, N., and Toyoda, M. (2017). A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 99–109, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Nguyen, T. and Chiang, D. (2018). Improving lexical choice in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers), pages 334–343, New Orleans, Louisiana. Association for Computational Linguistics.
- Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.
- Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Reiter, E. (2018). A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.
- Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units.

- In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.
- Stanojevic, M. and Sima'an, K. (2014). Fitting sentence level translation evaluation with many dense features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206, Doha, Qatar. Association for Computational Linguistics.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.
- Unicode Consortium (2020). The Unicode Standard. Unicode Consortium, Mountain View, CA.
- Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., and Uszkoreit, J. (2018). Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas* (Volume 1: Research Track), pages 193–199, Boston, MA. Association for Machine Translation in the Americas.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Wang, W., Peter, J.-T., Rosendahl, H., and Ney, H. (2016). CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.
- Williams, P., Sennrich, R., Nadejde, M., Huck, M., Haddow, B., and Bojar, O. (2016). Edinburgh’s statistical machine translation systems for WMT16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 399–410, Berlin, Germany. Association for Computational Linguistics.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *Computing Research Repository*, arXiv:1609.08144. version 2.