Indonesian Journal of Electrical Engineering and Computer Science
Vol. 34, No. 3, June 2024, pp. 1924~1934
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v34.i3.pp1924-1934

Automatic translation from English to Amazigh using transformer learning

Otman Maarouf[1], Abdelfatah Maarouf[1], Rachid El Ayachi[1], Mohamed Biniz[2]
[1]Department of Computer Science, Faculty of Science and Technology, Sultan Moulay Slimane University, Beni-Mellal, Morocco
[2]Department of Computer Science, Polydisciplinary Faculty Sultan Moulay Slimane University, Beni Mellal, Morocco

Article Info

Article history:

Received Dec 24, 2023
Revised Feb 12, 2024
Accepted Mar 4, 2024

Keywords:

Amazigh
Neural machine translation
OpenNMT
Statistical machine translation
Transformer

ABSTRACT

Due to the lack of parallel data, to our knowledge, no study has been conducted on the Amazigh-English language pair, despite the numerous machine translation studies completed between major European language pairs. We decided to utilize the neural machine translation (NMT) method on a parallel corpus of 137,322 sentences. The attention-based encoder-decoder architecture is used to construct statistical machine translation (SMT) models based on Moses, as well as NMT models using long short-term memory (LSTM), gated recurrent units (GRU), and transformers. Various outcomes were obtained for each strategy after several simulations: 80.7% accuracy was achieved using the statistical approach, 85.2% with the GRU model, 87.9% with the LSTM model, and 91.37% with the transformer.

Corresponding Author:

Otman Maarouf
Department of Computer Science, Faculty of Science and Technology, Sultan Moulay Slimane University
Av Med V, BP 591, Beni-Mellal 23000, Morocco
Email: maarouf.otman94@gmail.com

1. INTRODUCTION

Machine translation, a pivotal subfield in natural language processing, aims to bridge language barriers by enabling the automated translation of text between human languages [1]. This interdisciplinary field draws from mathematics, statistics, artificial intelligence, and linguistics to develop algorithms that translate one natural language into another [2]. The evolution of machine translation began in the mid-20th century and has progressed from rule-based systems to more advanced statistical and neural models [3].

Machine translation has made notable strides, yet it struggles with languages like Amazigh [4], which are less commonly represented in translation research. The intricate linguistic features of Amazigh, including its rich morphology and diverse dialects, present unique challenges [5]. These complexities are compounded by a scarcity of comprehensive parallel corpora, hindering the development of robust translation models [6]. Additionally, Amazigh's distinct syntactic and semantic structures require specialized attention, which existing translation models often fail to provide [7]. The lack of focused research and tailored resources for Amazigh further exacerbates the problem, leading to suboptimal translation outcomes. Addressing these issues is crucial for advancing machine translation capabilities for Amazigh and other underrepresented languages.

This paper presents a comparative study of three distinct approaches to Amazigh-English machine translation, each grounded in different technological advancements: statistical machine translation (SMT) [8], neural machine translation (NMT) [9], and transformer models [10]. Our research uniquely focuses on evaluating the efficacy of each model independently, with the aim of identifying the most effective technique for translating the complexities of the Amazigh language. We employ specialized preprocessing techniques and develop custom language models tailored to the intricacies of Amazigh, an approach not widely explored in previous research. This methodology allows for a comprehensive analysis and comparison of the strengths and limitations of SMT, NMT, and transformer models in the context of Amazigh-English translation.

The manuscript is organized into three main sections. The section 1 reviews existing studies on machine translation, highlighting their limitations in handling languages like Amazigh. The section 2 delves into the methodologies and results of SMT, examining its applicability to Amazigh. The section 3 focuses on NMT methods, including the use of transformer models, and discusses their effectiveness in translating Amazigh. The paper concludes by summarizing the findings and implications of our research, demonstrating the significance of our contributions to the field of machine translation.

2. RELATED WORKS

Numerous studies in the area of machine translation have been conducted. For example, in the Hindi language, we found [11] based on declension principles, suggesting a machine translation mechanism for Hindi to English. They also discuss various machine translation techniques. Arora *et al.* [12] suggested a multilingual machine translation program capable of translating into Hindi and Urdu. The system uses an artificial neural network and a translation rules-based technique. Neural networks are excellent for creating a translation rule-based machine translation system due to their effective pattern matching and capacity for learning from instances. Bhagavad Gita - the hymn of the Lord - is used as input data in the new corpus-based translation method for Sanskrit to Hindi proposed by [13]. In their research, deep neural networks are employed for training. After data analysis and processing, input data is supplied to the neural network, which then performs auto-tuning to improve the model.

The machine translation approach was also applied to the Arabic language. A rule-based machine translation system between Arabic and Arabic sign language (ArSL) is suggested by [14]. Their suggested approach translates ArSL text lexically and syntactically into Arabic by performing morphological and syntactic analysis. Almahairi *et al.* [15] compares NMT to a conventional phrase-based translation system for the problem of translating Arabic (Ar → En). They conduct in-depth comparisons of phrase-based and neural translation systems using various configurations for preparing Arabic scripts. The results demonstrate that both systems respond similarly to the correct preparation of the script. Brour and Benabbou [16] aims to create a translation system that can provide instantaneous assertions via a signing avatar. The method, which translates Arabic text into ArSL using a computer, will make it easier for those with hearing impairments to connect with others who have difficulty hearing and only know sign language.

The Amazigh language has also received attention from the machine translation approach, especially SMT. Taghbalout and Allah [17] provides a hybrid construction process for an Amazigh-to-English machine translation. The suggested system's design is based on both statistical-based machine translation (SBMT) and interlingua-based machine translation (IBMT) methodologies. As the initial stage in developing the linguistic resources needed by the UNL system to achieve translation [18] have started work on the Amazigh-UNL dictionary to create a machine translation system based on the universal networking language (UNL) for this language.

3. METHOD

3.1. Amazigh SMT

An approach to machine translation, known as SMT, generates translations based on statistical models whose parameters are derived from the analysis of large volumes of bilingual text corpora. This method operates by identifying patterns in the text that correlate with translations between the two languages. Essentially, SMT utilizes the frequency of phrases and their alignments in parallel texts to predict the most likely translation for a given segment of text [19].

3.1.1. Moses

To train statistical models of text translation from a source language to a target language, one can utilize Moses, a free SMT engine. Moses uses these models to automatically translate text into the target language by decoding fresh source language material. A parallel corpus, containing texts in both languages, is needed for the training, often consisting of pairs of manually translated sentences. Moses is distributed under the lesser general public license (LGPL) and is accessible as source code and executables for both Linux and Windows [19]. The EuroMatrix project which receives financing from the European Commission, is mostly responsible for supporting its development.

3.1.2. Statistical system work mechanism

A sentence must be converted into an Amazigh sentence. Given an English phrase E, we have a model $p(P|E)$ that calculates the conditional probability of any Amazigh sentence. To determine the parameters, we use the training corpus. The issue with machine translation is broken down into three steps:

- Language model $p(P)$, which can be the trigram model's estimate from any data. We do not require a parallel corpus for this language model.
- $p(P|E)$ translation model, this model must be trained on parallel corpora.
- Search for P maximizing the product $p(P|E)p(P)$ from these two models, we can develop a translation system.

3.1.3. Linguistic model

Consider breaking down an Amazigh phrase into the words $P = P_1, P_2, P_3 \dots . P_m$. The likelihood of E may thus be expressed as the sum of conditional probabilities as (1):

$$p(P) = \prod_{j=1}^{m} p(P_j | P_1, \dots .., P_{m-1}) \tag{1}$$

3.1.4. Translation model

The probability is calculated using the translation model. The word alignment from the source language to the target language is returned by the translation model, which estimates the parameters using parallel corpora.

$$p(P|E) = \frac{P(E,P)}{p(P)} = \frac{p(P)p(E \vee P)}{\sum p(P)p(E \vee P)} \tag{2}$$

Consider the following pair of aligned sentences (ⵢⵙⵙⵃⴽ ⴰⴷ ⴷ-ⵜⴳⵔⵎ ⵡⵄⵎⵇ ⵍⵙ ⴷⴳ ⵡⵖⵙⴰⵝ.) You(1) must(1) take(2)(3) his(5) age(4) into(6) account(7). Alignment of this parallel sentence: a = {1,2,3, 4, 5,6,7}.

Figure 1 illustrates the alignment between two sentences: one in English and the other in Amazigh, as demonstrated by a puzzle grid. Upon inspection, we can discern three types of alignments: 'many-to-one,' where multiple words or elements in one language correspond to a single element in another; 'one-to-many,' where a single element corresponds to multiple elements in the other language; and 'one-to-one,' where there is a direct equivalence between elements of both languages. These alignments are a common challenge in bilingual language processing and translation studies, where understanding the relationship between different linguistic elements is crucial. We will define the model for $p(P|E)$and $p(E|\text{a,P})$ giving as (3):

$$p(E|\text{a,P}) = p(a|P)p(E|a, P) \tag{3}$$

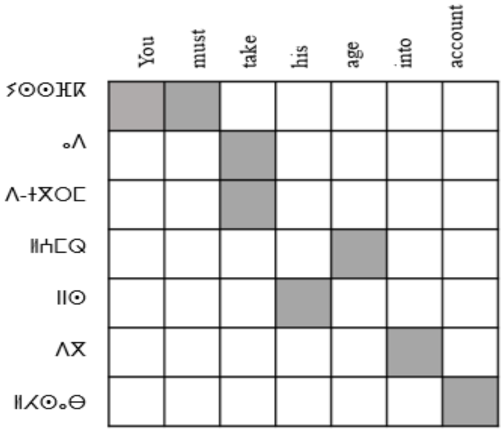


Figure 1. Alignment of two sentences (English-Amazigh)

3.1.5. Decoder

Decoding in SMT is a critical phase where the aim is to choose the target text that has the highest probability of being the correct translation of a given source text. This involves applying both the translation model, which provides alignment probabilities of phrases between the source and target languages, and the language model, which ensures linguistic fluency in the output. The decoder navigates through numerous possible translations, calculating their probabilities and selecting the one that maximizes the probability P(target|source). This process effectively handles language ambiguities and syntactical differences. The quality of the decoding process is contingent upon the robustness of the training data and the efficacy of the search algorithms used.

$$p(P|E)p(E,P) = argmax\, p\,(P)p(E|P) \tag{4}$$

The problem is finding a pair (P, E) that maximizes $p(\mathrm{P}, \mathrm{a}|E)$. Using Bayes' theorem, this is equivalent to finding (P, a) which maximizes:

$$p(\mathrm{P}, \mathrm{a}|E) = p(E, a|P)p(P)/p(E) \tag{5}$$

3.2. Neural machine translation

Automating translation can be achieved through NMT, a type of end-to-end learning. Rather than starting with a set of predetermined rules, NMT utilizes the software's neural network to encode and decode the original text [20]. Thus, NMT has the capability to address many of the issues associated with conventional sentence-based translation systems and has been demonstrated to provide translations of higher quality. One implementation of NMT employs the recurrent neural network (RNN), a powerful model for sequential data, for both the encoder and decoder [21]. The core components of NMT include an encoder that computes a representation of the source sentence and a decoder that produces one target word at a time to construct the conditional probability [22]. Figure 2 illustrates the encoder-decoder architecture used for the English-Amazigh NMT.

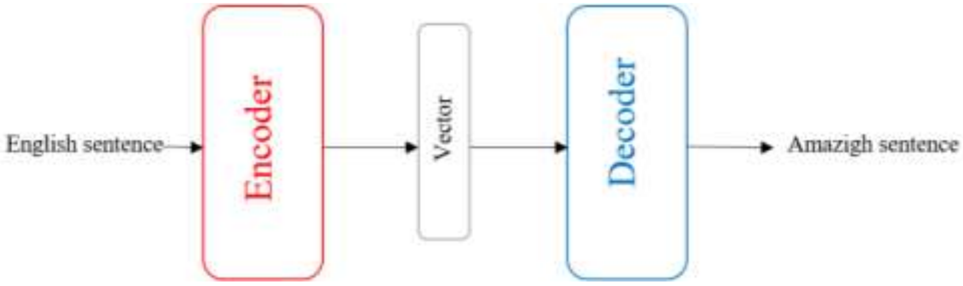


Figure 2. Encoder and decoder architecture

4. RESULTS AND DISCUSSION

In this section, we delve into the results of our research, presenting a thorough analysis and discussion of the findings. To ensure clarity and facilitate understanding, the results are showcased through a variety of formats, including figures, graphs, tables, and other visual aids. This approach not only makes the data more accessible but also allows for a more intuitive grasp of the implications and significance of our findings. The discussion is structured into several sub-sections, each focusing on different aspects of the study, from the comparison of SMT and NMT models to the implications of these results for the field of machine translation. This structured analysis aims to provide a comprehensive understanding of the research outcomes and their broader impact.

4.1. Statistical machine translation

The development of a SMT system begins with the collection and preparation of parallel corpora, ensuring the text is clean and normalized for consistency. Linguistic models are then created to understand and predict word sequences, while translation models map these sequences between the source and target languages. The Moses toolkit is employed to train these models, refining the translation process to produce accurate and fluent translations. This meticulous preparation and training is essential for building an effective SMT system that captures the nuances of both the source and target languages.

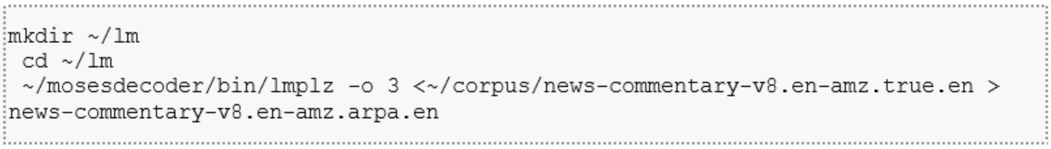
4.1.1. Corpus preparation

A significant barrier to the development of SMT systems is the absence of a parallel corpus. The parallel corpus consists of collections of sentences from both the source and target languages. For this study, an English-Amazigh parallel corpus comprising approximately 137,000 sentences was employed. It was collected from various websites. This corpus includes sentences from a wide range of domains. We utilized 20% of the corpus for testing and the remaining 80% for training.

4.1.2. Language model training

The Amazigh language serves as the target for a specialized translation system, requiring careful training with ample bilingual text to comprehend its unique linguistic features. The system's accuracy hinges on precise text alignment and command-line configurations, as detailed in the Moses user manual. These settings enable the model to produce coherent and culturally relevant translations, taking into account Amazigh's distinct phonetic and morphological characteristics. By meticulously calibrating the system, it aims to deliver high-quality translations that facilitate clear and meaningful communication for Amazigh speakers.

Figure 3 displays the command line sequence for training a language model using the Moses SMT framework. The commands outline the creation of a new directory, navigating to this directory and then executing the language model training script from the Moses toolkit with specific parameters and input files. This process is essential for developing a statistical model capable of translating or processing language data, in this case likely involving the English and Amazigh languages.



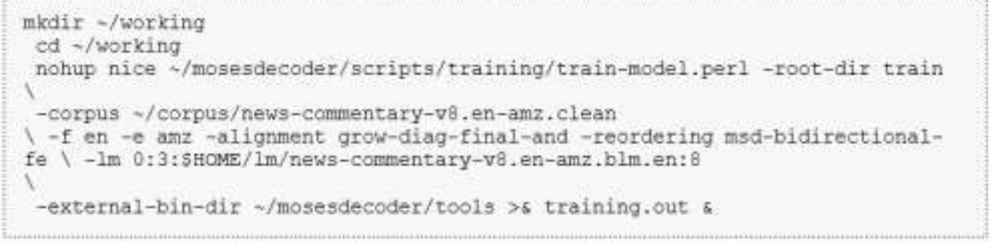
```
mkdir ~/lm
 cd ~/lm
 ~/mosesdecoder/bin/lmplz -o 3 <~/corpus/news-commentary-v8.en-amz.true.en >
news-commentary-v8.en-amz.arpa.en
```

Figure 3. Command to train language model

4.1.3. The translation system's training

The final phase in developing a SMT system is training the translation model with a specific language pair. In this phase, sentences from a parallel corpus are meticulously aligned using GIZA++. This process informs the construction of lexicalized reordering rules within the Moses toolkit configuration, ensuring that the output adheres to the grammatical structure of the target language. To facilitate this complex operation, all related data and scripts are systematically organized into a dedicated directory, serving as the command center for the training regime. The outcome is a nuanced translation model capable of handling both the literal and idiomatic complexities of the language pair, thereby enhancing the system's ability to deliver accurate and fluent translations.

Figure 4 illustrates the command-line instructions for training a translation system using the Moses decoder. The script initiates by creating and switching to a working directory. It then runs the 'train-model.perl' script from the Moses suite in a 'nohup' environment with the 'nice' command to regulate the job's priority. The command specifies the root directory for training, the corpus for both the source and target languages, alignment models, reordering models, and the language model with its order. It also sets the directory for external binary files and directs the output to a file, ensuring that the training process continues running in the background even if the session disconnects.



```
mkdir ~/working
 cd ~/working
 nohup nice ~/mosesdecoder/scripts/training/train-model.perl -root-dir train
\
 -corpus ~/corpus/news-commentary-v8.en-amz.clean
\ -f en -e amz -alignment grow-diag-final-and -reordering msd-bidirectional-
fe \ -lm 0:3:$HOME/lm/news-commentary-v8.en-amz.blm.en:8
\
 -external-bin-dir ~/mosesdecoder/tools >& training.out &
```

Figure 4. Command line to train translation system

4.1.4. Testing

Now we run moses with the following command: “~/Amaz_moses/bin/moses -f \~/work/mert-work/moses.ini” We can type a sentence in English and get the output in Amazigh. We use “echo” to display the output like this: “ Echo "i can go." ” “~/ Amaz_moses/bin/moses -f ~/work/mert-work/moses.ini” This will give the output: “ⵣⵎⵔⵖ ⴰⴷ ⵇⵓⵢⵖ.”. Now, we can assess the precision of our suggested system. This test set evaluates the decoder, after which we execute the BLEU scripts on it. We use a number of datasets, each containing 10 to 100 phrases, for the studies. We track both successful and unsuccessful sentence translations. Figure 5 visually shows these results. The bilingual evaluation understudy (BLEU) was used to assess the outcome. The system's output translation quality is evaluated using the BLEU toolbox. The average degree of accuracy is 80.7%.

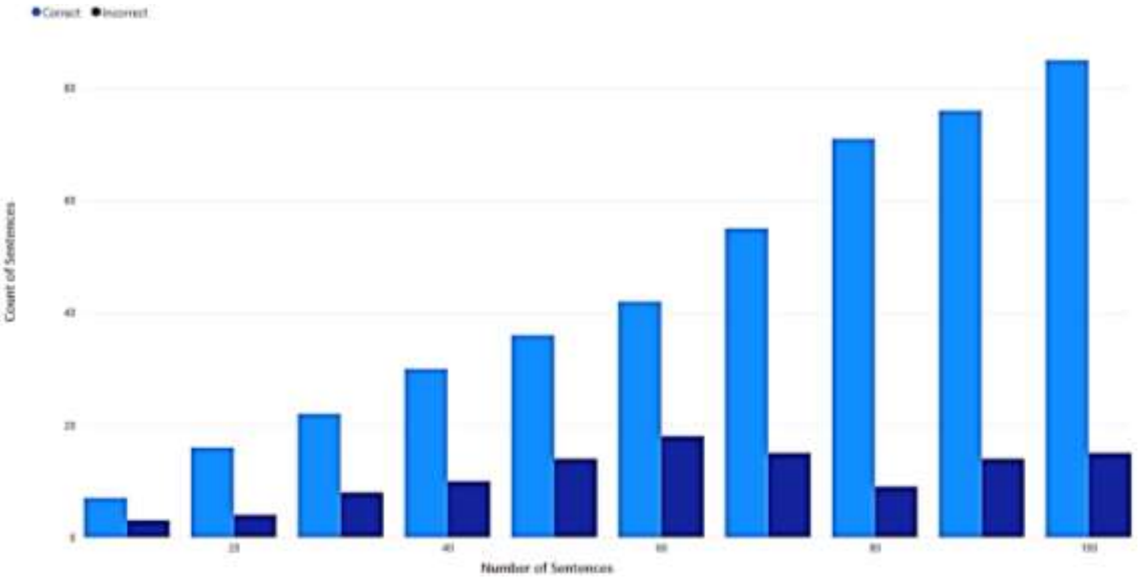


Figure 5. Graphical analysis for sentence observation

4.2. Comparison of the performance NMT models

In this section, we will present a detailed comparison of the performance and efficiency of three leading NMT models: the gated recurrent unit (GRU), long short-term memory (LSTM), and the Transformer. Each model represents a unique approach to tackling the complexities of machine translation, leveraging different architectural innovations to improve translation quality and processing speed.

4.2.1. Data preparation and text pre-processing

A. Download and prepare data

A parallel corpus of 137,322 sentences is used. This corpus includes fresh Latin sentences that are commonly used in everyday life, as well as Tifinagh script phrases scraped from the web. The corpus was organized and refined through a cleaning process. After these procedures, a manual validation was carried out to ensure there were no errors.

Here are the procedures to follow after preparing the data and importing it from the Hard Disk Drive as text files:

- Give each phrase a START and END token.
- Remove any unnecessary special characters from the sentences.
- Create a word index and a reverse word index (word to id and id to word dictionary mapping).
- Extend each statement to its full potential.

We efficiently built a "tf.data.Dataset" of strings from these string arrays.

B. Text pre-processing-normalization

A model known as the NMT model can be exported as a tf.saved_model. For this model, tf.string inputs and outputs are required, and all text processing is carried out within the model itself. The model utilizes a limited vocabulary for handling multilingual content. Consequently, normalizing the provided text is crucial. The first step involves the separation of accented letters and their replacement with their American standard code for information interchange (ASCII) counterparts, achieved through Unicode normalization. One such Unicode normalization operation can be found in the TensorFlow Text package.

4.2.2. Simple NMT

A. NMT with TensorFlow model

A corpus comprising 137,322 parallel phrases, along with six layers of encoders and six layers of decoders, was utilized to develop this approach. The model underwent testing on Google Colab, a popular cloud-based platform that facilitates machine learning experiments. To enhance the model's accuracy, specific values or patterns, which are not directly deducible from the data, were estimated externally. These estimations, known as hyperparameters, play a crucial role in fine-tuning the model's performance. They include settings such as learning rate, batch size, and the number of epochs, and are essential for optimizing the model's ability to learn from the data and make accurate translations. The careful selection and adjustment of these hyperparameters are key to achieving the desired level of translation quality in the machine translation system.

The various results obtained from several assessment criteria following extensive simulations are displayed in Figure 6. The resulting average BLEU score is 44.76. The graph clearly demonstrates that the BLEU score decreases as the word error rate increases and rises as the word error rate decreases. This is because a low word error rate indicates high translation quality, and consequently, a high word error rate corresponds to a lower BLEU score.

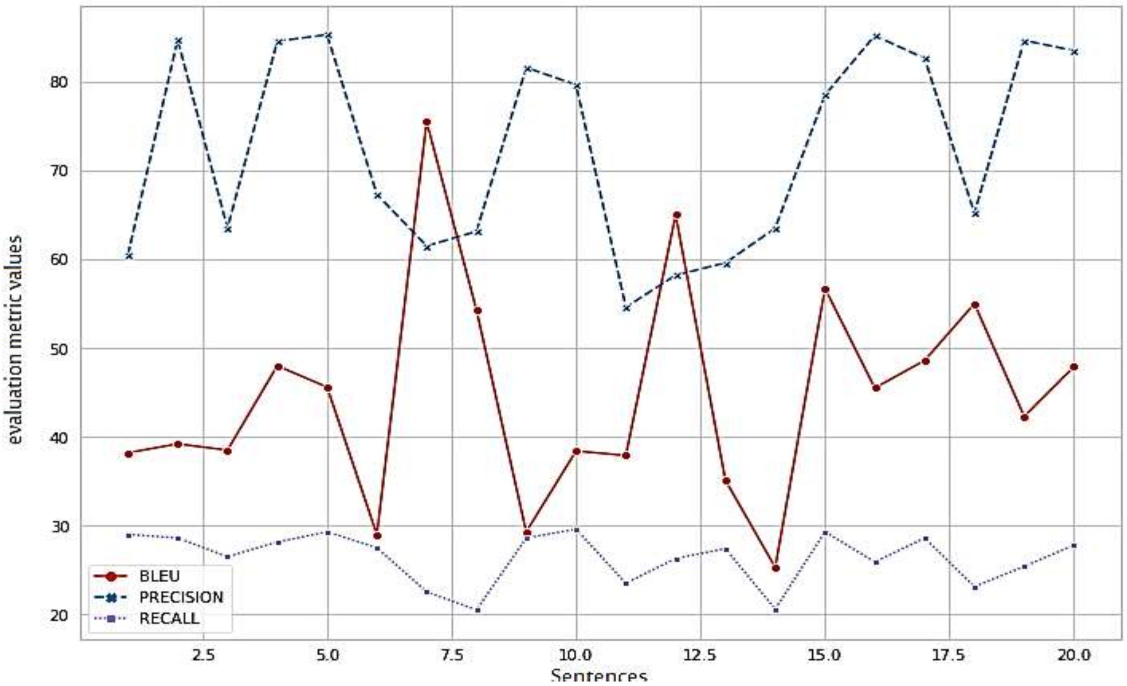


Figure 6. Graphical representation of the BLEU, PRECISION, and RECALL measurements of the tensorflow model

B. NMT with PyTorch (OpenNMT) model

To build the NMT model and translate text from English to Amazigh, the OpenNMT system, an open-source toolkit for NMT, was utilized. The performance of the NMT system is greatly influenced by the preprocessing of Tifinagh and English characters. Sentences are separated and aligned manually. After extensive testing, the maximum length of the source and target sequences was set to 44, the maximum batch size for training and validation was established at 128, and the dropout rate was set at 0.3 for the LSTM RNN types with Adam optimization. Other parameters are set to their default values. The system records the model for every 10 epochs and then calculates the accuracy and perplexity (PPL) 10 times for each model. Perplexity refers to how easily a probability distribution (the model) can predict the next sequence. The translation model is considered effective at predicting/translating the test set if the perplexity is low.

Figure 7 displays the evaluation of the LSTM-based NMT model. In this figure, 'Val ppl' stands for 'Validation Perplexity,' 'Val Acc' denotes 'Validation Accuracy,' 'PPL' represents the 'Average Prediction Score,' and 'Accuracy' indicates the 'Prediction Perplexity.' These metrics collectively provide insights into the model's performance. The results demonstrate that the LSTM-based NMT model performs effectively, achieving a high validation accuracy score of 87.9%. This indicates a strong ability of the model to accurately translate text while maintaining a low level of perplexity, reflecting its efficiency in dealing with the complexity of the language structure in the translation process. The low prediction perplexity suggests that the model is capable of generating translations with high confidence, and the high validation accuracy highlights its effectiveness in correctly translating a wide range of sentences.

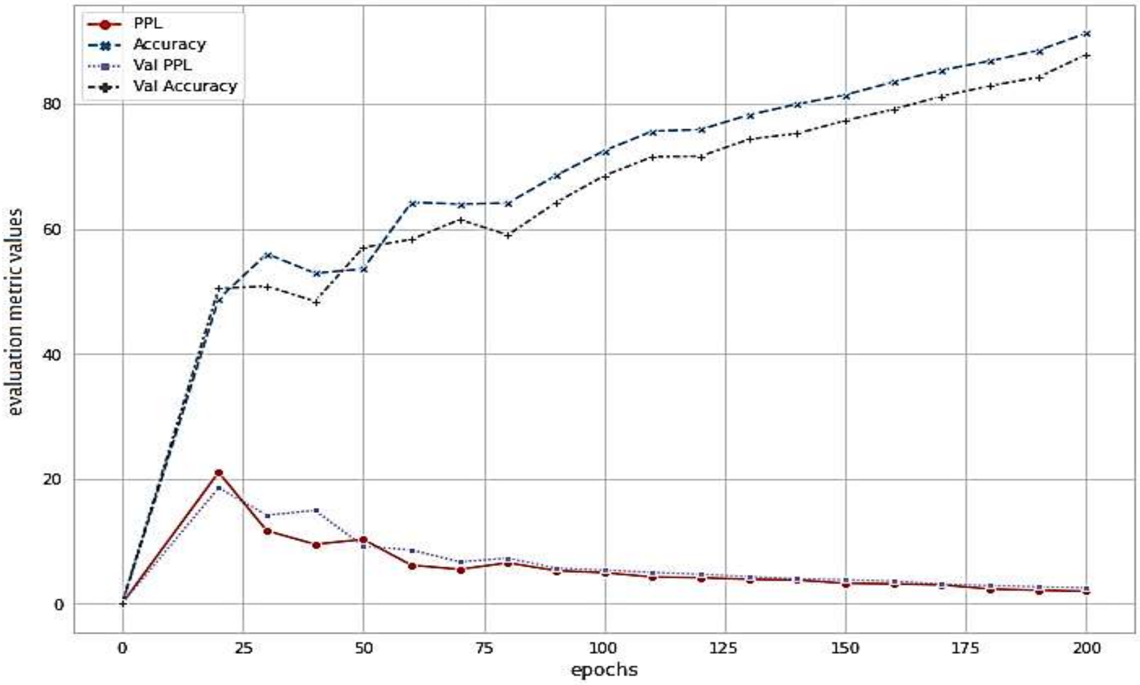


Figure 7. Graphical representation of the PPL, accuracy measurements of the OPNNMT model

4.2.3. NMT with transformer models

The transformer introduces an innovative approach to machine translation, leveraging the attention mechanism to efficiently capture dependencies. This method significantly reduces computational and training time compared to previous models. Google's research team has evaluated its performance, noting that the Transformer has achieved the highest BLEU scores in machine translation tasks, marking a substantial advancement in the field.

The transformer architecture has been successful in covering a large number of language pairs with high accuracy in machine translation (MT) tasks. This includes models such as bidirectional and auto-regressive transformers (BART) [23] and multilingual BART (mBART) [24]. Furthermore, the multilingual text-to-text transfer transformer (mT5) [25] performs well with an even larger set of languages, many of which are considered low-resource. We have built an MT system based on the transformer architecture. Since the transformer is the state-of-the-art architecture for MT and our system has been trained on considerable amounts of domain-specific data, the translations produced by our system are representative of the high quality that can be achieved in today's MT systems for translation from English to Amazigh. Our implementation of the transformer architecture uniquely adapts it for the Amazigh language, taking into account its specific linguistic features and challenges, which sets our system apart in the field of machine translation.

Figure 8 shows the evolution of performance metrics after every 10 epochs of training. It is evident that the inclusion of syntactic information brings significant benefits to most of the adopted metrics. We observe that the training accuracy increases from 65.25% to 94.56%, and the validation accuracy increases from 64.25% to 91.3%.

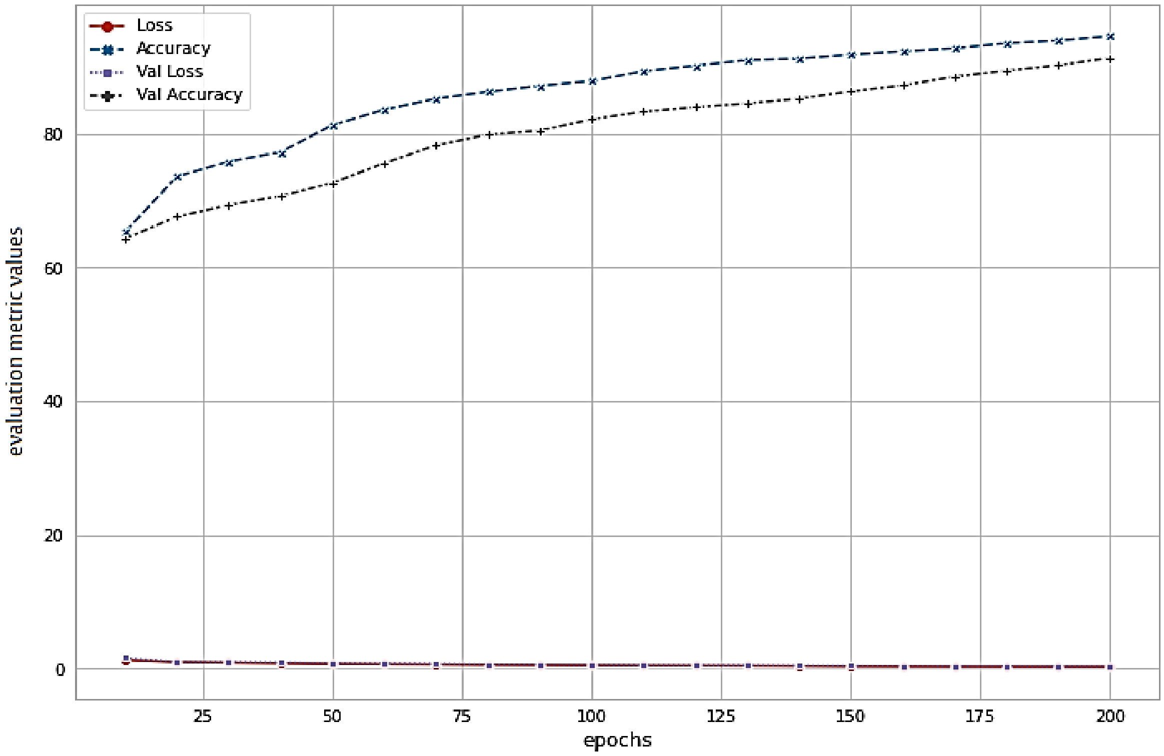


Figure 8. Graphical representation of the PPL, accuracy measurements of the OPNNMT model

4.3. Comparison of translation models

We developed and tested a GRU cell-based multilingual NMT system. The hyperparameters utilized for each of our models are listed in Table 1. The LSTM and transformer models are trained using the open-source toolkit OpenNMT-tf3, while the GRU experiments are carried out using the NMT toolbox Nematus [26]. The following are the settings for the toolkits' training and inference hyperparameters.

Table 1. Hyper-parameters used to train GRU, LSTM and transformer models

Encoder/ decoder	Embiding size	Hidden units	Encoder depth	Decoder depth	Batch size	Accuracy%
GRU	256	1024	2	2	64	85.2
LSTM	1024	1024	2	2	128	87.9
Transformer	512	512	6	6	64	91.3

In this study, we have examined the comparative effectiveness and efficiency of different neural network architectures, namely transformer, LSTM, and GRU models, in a specific task, alongside the traditional SMT model. Our findings clearly indicate that the transformer model outperforms its counterparts, achieving an impressive 91.3% accuracy in validation, which is significantly higher than that of the LSTM (87.9%), GRU (85.2%), and notably surpassing the SMT model's accuracy of 80.7%. This superiority in performance can be attributed to the transformer's innovative architecture that facilitates faster training speeds, which were observed to be 30% quicker than the GRU model and 45% faster than the LSTM model, a benefit largely due to the efficient utilization of GPU technology. Despite these promising results, it is essential to acknowledge the limitations of our research, particularly in terms of the scope of datasets used, which might not fully represent all potential use cases. Therefore, future research should focus on exploring these models across a broader spectrum of tasks and conditions to validate the generalizability of our findings. Additionally, further investigation into optimizing the performance of LSTM and GRU models could provide more competitive alternatives to the transformer model, especially in scenarios where resource constraints are a critical factor.

5. CONCLUSION

NMT represents a groundbreaking shift in the field of machine translation research. In this study, we developed an LSTM-based deep encoding-decoding model for translating English to Amazigh. Our approach incorporated the Bahdanau attention mechanism, utilizing a parallel Amazigh-English corpus of 137,322 sentences. To assess the effectiveness of our proposed system, we employed various automatic evaluation measures, including BLEU, F-measure, Accuracy, and Perplexity. Extensive simulations yielded distinct results for each technique: 80.7% accuracy using the static method, 85.2% with the GRU model, 87.9% via the LSTM model, and a notable 91.3% accuracy achieved with the transformer model.

REFERENCES

[1] M. A. Kumar, B. Premjith, S. Singh, S. Rajendran, and K. P. Soman, "An overview of the shared task on machine translation in indian languages (MTIL) – 2017," *Journal of Intelligent Systems*, vol. 28, no. 3, p. 455-464, 2019, doi: 10.1515/jisys-2018-0024.

[2] S. Jindal, V. Goyal, and J. S. Bhullar, "English to Punjabi statistical machine translation using moses (Corpus Based)," *Journal of Statistics and Management Systems,* vol. 21, no. 4, p. 553-560, 2018, doi: 10.1080/09720510.2018.1471265.

[3] A. Waibel, A. N. Jain, A. E. McNair, H. Saito, A. G. Hauptmann, and J. Tebelskis, "JANUS: a speech-to-speech translation system using connectionist and symbolic processing strategies," in *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ont., Canada: IEEE, 1991, p. 793-796 vol. 2. doi: 10.1109/ICASSP.1991.150456.

[4] R. Ammari, L. Zenkouar, and M. Outahajala, "System of analysis and generation for amazigh nominal morphology based on xerox's finite-state transducer," *Procedia Computer Science,* vol. 117, p. 217-224, 2017, doi: 10.1016/j.procs.2017.10.112.

[5] K. E. Gajoui, F. A. Allah, and M. Oumsis, "Diacritical language OCR based on neural network: case of amazigh language," *Procedia Computer Science,* vol. 73, p. 298-305, 2015, doi: 10.1016/j.procs.2015.12.035.

[6] I. Taghbalout, F. A. Allah, and M. E. Marraki, "Amazigh representation in the UNL framework: resource implementation," *Procedia Computer Science,* vol. 73, p. 234-241, 2015, doi: 10.1016/j.procs.2015.12.023.

[7] Y. Chaabi and F. A. Allah, "Amazigh spell checker using Damerau-Levenshtein algorithm and N-gram," *Journal of King Saud University - Computer and Information Sciences,* vol. 34, no. 8, p. 6116-6124, 2022, doi: 10.1016/j.jksuci.2021.07.015.

[8] E. Satir and H. Bulut, "Preventing translation quality deterioration caused by beam search decoding in neural machine translation using statistical machine translation," *Information Sciences,* vol. 581, p. 791-807, Déc. 2021, doi: 10.1016/j.ins.2021.10.006.

[9] A. V. Hujon, T. D. Singh, and K. Amitab, "Neural machine translation systems for English to Khasi: A case study of an Austroasiatic language," *Expert Systems with Applications,* vol. 238, p. 121813, Mars 2024, doi: 10.1016/j.eswa.2023.121813.

[10] T. Tian, C. Song, J. Ting, and H. Huang, "A french-to-english machine translation model using transformer network," *Procedia Computer Science,* vol. 199, p. 1438-1443, 2022, doi: 10.1016/j.procs.2022.01.182.

[11] J. Nair, K. A. Krishnan, and R. Deetha, "An efficient English to Hindi machine translation system using hybrid mechanism," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, India: IEEE, p. 2109-2113, Sept. 2016, doi: 10.1109/ICACCI.2016.7732363.

[12] A. Arora, N. Venkateswaran, and N. Schneider, "Hindi-Urdu adposition and case supersenses v1.0," *arXiv preprint arXiv:2103.01399,* 2021, doi: 10.48550/ARXIV.2103.01399.

[13] M. Singh, R. Kumar, and I. Chana, "Corpus based machine translation system with deep neural network for sanskrit to Hindi translation," *Procedia Computer Science,* vol. 167, p. 2534-2544, 2020, doi: 10.1016/j.procs.2020.03.306.

[14] H. Luqman and S. A. Mahmoud, "A machine translation system from Arabic sign language to Arabic," *Universal Access in the Information Society*, vol. 19, no 4, p. 891-904, Nov. 2020, doi: 10.1007/s10209-019-00695-6.

[15] A. Almahairi, K. Cho, N. Habash, and A. Courville, "First result on Arabic neural machine translation," *arXiv preprint arXiv:1606.02680,* 2016, doi: 10.48550/ARXIV.1606.02680.

[16] M. Brour and A. Benabbou, "ATLASLang MTS 1: Arabic text language into arabic sign language machine translation system," *Procedia Computer Science,* vol. 148, p. 236-245, 2019, doi: 10.1016/j.procs.2019.01.066.

[17] I. Taghbalout and F. A. Allah, "A hybrid approach for amazigh-english machine translation," in *Proceedings of the 7th International Conference on Software Engineering and New Technologies*, Hammamet Tunisia: ACM, Déc. 2018, p. 1-6, doi: 10.1145/3330089.3330113.

[18] I. Taghbalout, F. A. Allah, and M. E. Marraki, "Towards UNL-based machine translation for Moroccan Amazigh language," *International Journal of Computational Science and Engineering,* vol. 17, no. 1, p. 43-54, 2018, doi: 10.1504/IJCSE.2018.094418.

[19] X. Wang, Z. Lu, Z. Tu, H. Li, D. Xiong, and M. Zhang, "Neural machine translation advised by statistical machine translation," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 31, no. 1, 2017, doi: 10.1609/aaai.v31i1.10975.

[20] S. Ranathunga, E.-S. A. Lee, M. P. Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural machine translation for low-resource languages: a survey," *ACM Computing Surveys,* vol. 55, no. 11, p. 1-37, Nov. 2023, doi: 10.1145/3567592.

[21] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," *ArXiv Prepr. ArXiv14108206*, 2014, doi: 10.48550/arXiv.1410.8206.

[22] F. Hieber *et al.*, "Sockeye 3: fast neural machine translation with PyTorch," *arXiv preprint arXiv:2207.05851,* 2022, doi: 10.48550/ARXIV.2207.05851.

[23] M. Lewis *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *ArXiv Prepr. ArXiv191013461*, 2019, doi: 10.48550/arXiv.1910.13461.

[24] Y. Liu *et al.*, "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics,* vol. 8, p. 726-742, 2020, doi: 10.1162/tacl_a_00343.

[25] L. Xue *et al.*, "mT5: A massively multilingual pre-trained text-to-text transformer," *ArXiv Prepr. ArXiv201011934*, 2020, doi: 10.48550/arXiv.2010.11934.

[26] R. Bawden, R. Sennrich, A. Birch, and B. Haddow, "Evaluating discourse phenomena in neural machine translation," *ArXiv Prepr. ArXiv171100513*, 2017, doi: 10.48550/arXiv.1711.00513.

BIOGRAPHIES OF AUTHORS

Otman Maarouf received his master's degree in business intelligence in 2018 and a Ph.D. degree in computer science from the Faculty of Science and Technology, University Sultan Moulay Sliman Beni-Mellal. His research activities are located in the area of natural language processing specifically; it deals with part of speech, named entity recognition, and limatization-steaming of the Amazigh language. He can be contacted at email: Maarouf.otman94@gmail.com.

Abdelfatah Maarouf received his master's degree in business intelligence in 2022 and a Ph.D. student in computer science from the Faculty of Science and Technology, University Sultan Moulay Sliman Beni-Mellal. His research activities are located in the area of natural language processing and computer vision. He can be contacted at email: abdelfatah0maarouf@gmail.com.

Rachid El Ayachi obtained a degree in Master of Informatic Telecom and Multimedia (ITM) in 2006 from the Faculty of Sciences, Mohammed V University (Morocco) and a Ph.D degree in computer science from the Faculty of Science and Technology, Sultan Moulay Slimane University (Morocco). He is currently a member of laboratory TIAD and a professor at the Faculty of Science and Technology, Sultan Moulay Slimane University, Morocco. His research focuses on image processing, pattern recognition, machine learning, and semantic web. He can be contacted at email: rachid.elayachi@usms.ma.

Mohamed Biniz received his master's degree in business intelligence in 2014 and Ph.D degree in computer science in 2018 from the Faculty of Science and Technology, University SultanMoulay Sliman Beni-Mellal. He is a professor at polydisciplinary faculty University Sultan My Slimane Beni Mellal morocco. His research activities are located in the area of the semantic web engineering and deep learning specifically, it deals with the research question of the evolution of ontology, big data, natural language processing, and dynamic programming. He can be contacted at email: mohamedbiniz@gmail.com.