

Journal of Digital Science



ISSN 2686-8296

Volume 2 Issue 1

May 2020

© Institute of Certified Specialists

CONTENTS

A Review and evaluation of Machine Translation methods for Lumasaaba	3
Peter Nabende	
Neural Network Model for Assessing the Physical and Mechanical Properties of a Metal Material Based on Deep Learning	18
Andrei Kliuev, Roman Klestov, Valerii Stolbov	
Multifactorial model of adverse events and medical safety management	29
Yury Voskanyan, Irina Shikina, Olga Andreeva, Fedor Kidalov, David Davidov	
The influence of FDI on sustainable economic development of Ukraine in terms of global digitization	40
Svitlana Tkalenko, Natalya Sukurova	
Development and financial support of tourism exports in the digital economy	54
Aleksandr Gudkov, Elena Dedkova	
Financial reporting and climate-related disclosures	67
Olga Efimova, Olga Rozhnova	
Digital universities in Russia: digitization with extra speed	76
Darya Rozhkova, Nadezhda Rozhkova, Uliana Blinova	
Model of the communication process in a context of reading in French first language and French foreign language	82
Julia Belyasova, Michel Martin	

A Review and evaluation of Machine Translation methods for Lumasaaba

Peter Nabende¹[0000-0003-2141-7940]

¹ Department of Information Systems, Makerere University,
P. O. Box 7062, Kampala, Uganda

https://doi.org/10.33847/2686-8296.2.1_1

Received 13.03.2020/Revised 10.04.2020/Accepted 11.05.2020/Published 28.05.2020

Abstract. Natural Language Processing for under-resourced languages is now a mainstream research area. However, there are limited studies on Natural Language Processing applications for many indigenous East African languages. As a contribution to covering the current gap of knowledge, this paper focuses on evaluating the application of well-established machine translation methods for one heavily under-resourced indigenous East African language called Lumasaaba. Specifically, we review the most common machine translation methods in the context of Lumasaaba including both rule-based and data-driven methods. Then we apply a state of the art data-driven machine translation method to learn models for automating translation between Lumasaaba and English using a very limited data set of parallel sentences. Automatic evaluation results show that a transformer-based Neural Machine Translation model architecture leads to consistently better BLEU scores than the recurrent neural network-based models. Moreover, the automatically generated translations can be comprehended to a reasonable extent and are usually associated with the source language input.

Keywords: machine translation, Lumasaaba, data-driven machine translation, phrase-based statistical machine translation, Neural machine translation.

1. Introduction

There is a growing need to provide Natural Language Processing (NLP) resources and applications for under-resourced languages. Machine Translation (MT) is one NLP application that has been of greater use and impact in the exchange of information across languages. For example, different forms of MT have been incorporated in various types of Web-based applications to enable access to information in more than one natural language. The access to MT not only benefits the consumer but also the provider of the service, even when in some cases it is free. Machine translation comprises several methods under two major approaches: rule-based MT and data-driven MT. Rule-based MT approaches require sufficient linguistic input while data-driven approaches require lots of parallel sentences. Despite the general progress in MT using methods from the different approaches, many under-resourced languages are yet to be involved and hence lack any knowledge concerning the development of MT and resulting benefits. Since natural languages usually have distinct inherent properties, there is always a need to study them separately in the context of NLP so as to guide the development of effective solutions. In this regard, we focus on one heavily under-resourced East African language called Lumasaaba which has not yet been involved in any forms of MT. The first contribution of the paper is a review of the main MT methods in the context of automating translation between Lumasaaba and English. The second contribution of the paper is an application and evaluation of state-of-the-art approach to MT called Neural Machine translation using a limited corpus of parallel sentences from the Bible. The rest of the paper is organized as follows: In section 2, we present an overview of the Lumasaaba language; in section 3, we

present a review of the different MT methods in the context of Lumasaaba; in section 4, we present the data and methodology for evaluating a state of the art approach in Neural Machine Translation in an experimental Lumasaaba-to-English MT task; in section 5, we present the Neural Machine Translation evaluation results; section 6 concludes the paper with pointers to future work.

2. The Lumasaaba Language

Lumasaaba is regarded as a Bantu language under the Niger Congo language family. The language is mostly spoken in the Eastern part of Uganda around the western side of Mt. Elgon. The language has a number of dialects under two main categories [1]: the 'southern' dialects (for example: Lubuya, Lusoba, Lukiende, etc.) and the 'northern' dialects (for example Ludadiri, Luwalasi, Lufumbo, etc.). There are clear differences in both pronunciations and vocabularies between the two categories of dialects. The differences warrant a separate treatment of the two main categories of dialects with regard to MT. In this paper, we focus on a representation that is mostly associated with the 'southern' dialects. A reasonable amount of monolingual text and bilingual text (with English) has been generated can be used for MT study.

The representation of the 'southern' dialects just like a number of African Bantu languages uses the modern Latin alphabet except two characters (Xx and Qq) but with another character ŋ 'ng'. The language recognizes the usual common parts of speech although some of them are very limited. These are some example words [2]: nouns (umulimi 'farmer', Wele 'God') , adjectives (buli 'each', nanu 'who'), pronouns (ise 'I', iwe 'you', ifwe 'we'), verbs (bona 'see', soma 'read') , adverbs (hano 'here', kindi 'in another way') prepositions (ku 'on', ni 'with'), conjunctions (ne 'but', oba 'or', and interjections (aso! 'now you see!').

3. Review of MT methods in the context of Lumasaaba

This section reviews the two main MT approaches (Rule-based MT and Data-driven MT (Example-based, PBSMT and NMT) while illustrating their application to translation between Lumasaaba and English. We identify opportunities and requirements of implementing MT involving Lumasaaba for some of the methods in each approach.

3.1 Rule-based Machine Translation

The Rule-based MT approach is regarded as the first approach in the field of machine translation. As the name suggests, the Rule-based MT approach mainly involves the use of linguistic rules of various kinds including rules for syntax, lexical, lexical transfer, syntactic generation, morphological and semantic analysis [3]. The main principle of Rule-based MT is based on linking the structure of a 'source language' sentence with the structure of a 'target language' sentence. The following example illustrates the general application of the approach to English-to-Lumasaaba translation as we point out what is required to achieve each of the steps. Consider this sentence in English (source): *The Minister said that "this project will increase food"*. At a basic level, we need the following resources to get a Lumasaaba translation: a dictionary for mapping each English word to a suitable Lumasaaba word; rules for representing regular English structure; rules for representing regular Lumasaaba structure; and rules for relating the English and Lumasaaba structures together. Based on these requirements, the following steps follow:

Step 1: identify English part of speech information for each English word. In this step, English part of speech taggers can be used and different approaches are possible

including rule-based tagging, stochastic tagging, and transformation-based tagging ([4][5]). Using the Penn Treebank tag set [6], we get the following annotation for the English sentence:

*The=DT Minister=NN said=VBD that=IN this=DT project=NN will=MD.VB
increase=VB food=NN.*

Translation in the direction of Lumasaaba-to-English will also require part of speech information for a Lumasaaba 'source' sentence. This implies the need for a Lumasaaba part of speech tagger.

Step 2: Get syntactic information for the source sentence and parse it. Different grammatical formalisms and parsing methods can be used to get syntactic information. Using the Link parser [7] and the Penn Treebank tag set leads to the following syntactic information for the example:

(S (NP The Minister) (VP said (SBAR that (S (NP this project) (VP will (VP increase (NP food)))))))).

Step 3: Translate English words into Lumasaaba. Assuming the same part of speech for English, we will get the following:

The (category = DT) → (), Minister (category = NN) → Minista (category = NN), said (category = VBD) → alomele (category = VBD), that (category = SBAR) → ari (category = SBAR), this (category = DT) → yino (category = DT), project (category = NN) → i polojekiti (category = NN), will (category = MD.VB) → (), increase (category = VB) → ongela (category = VB), food (category = NN) → bilyo (category = NN).

Step 4: Map dictionary entries into appropriate inflected forms to have the following final generation:

'target language sentence': Minista alomele ari "I polojekiti yino inongela bilyo."

There are currently very limited resources for the Lumasaaba language to facilitate the steps above. Rule-based MT is still very necessary either as a standalone approach or in combination with other state-of-the-art MT methods for achieving high quality translations in various applications. So the development of linguistically motivated resources (such as part of speech taggers, parsers, and bilingual dictionaries) for low-resourced languages should also be prioritized. If resources were available, it would be interesting to explore currently active Rule-based frameworks such as Aperiium [8] and Grammatical Framework [9].

3.2 Data-driven machine translation

The field of machine translation is currently dominated by data-driven (or corpus-based approaches) [10] which can be categorized into three: Example-based machine translation (EBMT), statistical machine translation (SMT), and Neural Machine Translation (NMT). Data-driven approaches require a parallel corpus (which is at least sentence aligned) between two languages which is used to either build a knowledge base or translation models for automating translation.

3.2.1 Example-based Machine Translation

Example-based MT was proposed as an alternative to Rule-based MT and its origins are attributed to a paper presented by Makoto Nagao at a 1981 conference

and which was later published in 1984. The idea behind the EBMT approach is captured in Nagao's much quoted statement [11]:

Man does not translate a simple sentence by doing deep linguistic analysis, rather, Man does translation first, by properly decomposing an input sentence into certain fragmental phrases ..., then by translating these phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference.

EBMT is generally composed of three stages: search for examples and matching, extraction and retrieval of examples, and recombination [10]. The first stage takes a source language sentence and tries to find examples in the knowledge base which closely match it. For effective results, the search process is usually linguistically motivated. For matching, the input sentence is parsed into segments of a certain granularity; then each segment is matched with example segments from the source language section of the knowledge base at the same level of granularity; the example translations from the target language side of the knowledge base are also retrieved for each source language example segment. The third stage involves determining an alignment of the retrieved segments so as to produce a grammatical output in the target language.

Although EBMT is considered to be data-driven, the techniques that have been used at different stages of the process come in several flavors ranging from those that rely on less to those that rely on more linguistic information. The effort in developing the knowledge base for EBMT depends on the features that the examples should have. Of course, manual construction of the knowledge base is possible and can guarantee high quality results in specific cases but will take a lot of time. Automatic induction of examples from a parallel corpus for the knowledge base can be used to overcome the limitation of manual construction, but at the expense of quality. There are also other issues to address when using EBMT [12] including: determining the optimal level of granularity, determining the limit at which more examples may not improve the quality of translations; determining suitable examples; how to store examples for efficient searching especially when lexical, syntactic and other information is included; generalization of examples. Where linguistic information is required, the resources that facilitate Rule-based MT should suffice for EBMT as well. In this case, we lack the same resources for Lumasaaba as mentioned in the previous section. Alternatively, statistical information about the examples can be used; in this case, the knowledge base stores precomputed statistical parameters associated with bilingual word pairings and the "translation model" [10]. A target language model which gives the probabilities of target language sentences is also precomputed. Here, the EBMT approach becomes an SMT approach. There are a number of open source tools that can facilitate the development of an EBMT system as long as we have a sentence aligned Lumasaaba-English parallel corpus with additional linguistic information. We have established that the parallel corpus in its basic form (with just parallel sentences) is not enough for EBMT and the resources required for specifying additional linguistic information are currently non-existent for Lumasaaba.

3.2.2 Statistical Machine Translation

Statistical Machine Translation constitutes a considerable amount of the near past in the field of MT. This is mainly attributed to the relative ease with which SMT tools can be configured to work with available parallel corpora. The SMT approach has its beginnings in IBM's candid project in the late 1980s and early 1990s and the methods

and principles defined by Brown et al. [13] underpin other SMT approaches. The most popular of these approaches is Phrase-based Statistical Machine Translation (PBSMT) which considers a sequence of words (or phrases) as well as single words as the fundamental units of translation. Just before 2016, the best performing MT systems for language pairs with adequate data were based on the PBSMT approach and some Web-based MT systems were using this approach. In the following paragraphs, we describe the PBSMT approach in the context of Lumasaaba-to-English translation. Consider the following Lumasaaba sentence:

n'unengekhareengekhele shiseenge she khukonamo shitweela.

Figure 1 illustrates Lumasaaba-to-English translation using the PBSMT approach. In the figure, the Lumasaaba sentence is first broken up into phrases which are then translated using a phrase translation model; the phrase translations are then reordered according to a reordering model to generate the final target sentence. The best English translation (e_{best}) given the Lumasaaba input sentence l is defined as [14]:

$$e_{best} = \operatorname{argmax}_e p(l|e) \times p_{LM}(e) \quad (1)$$

where $p_{LM}(e)$ is obtained from an English language model. The language model measures the likelihood or fluency of a sequence of words in the English language and is also used to choose the most likely sequence from a set of propositions. Several natural language modeling methods and techniques can be used including n-gram methods, neural network-based methods, dynamic Bayesian networks, and other statistical and probabilistic techniques. Most of these methods and techniques are language independent. The other component of equation 1, $p(l|e)$ is mainly associated with the actual translation, and for PBSMT, it is decomposed further into translation and reordering components as follows [14]:

$$p(l_1|e_1^n) = \prod_{i=1}^n \phi(l_i|e_i) \times d(\operatorname{start}_i - \operatorname{end}_{i-1} - 1) \quad (2)$$

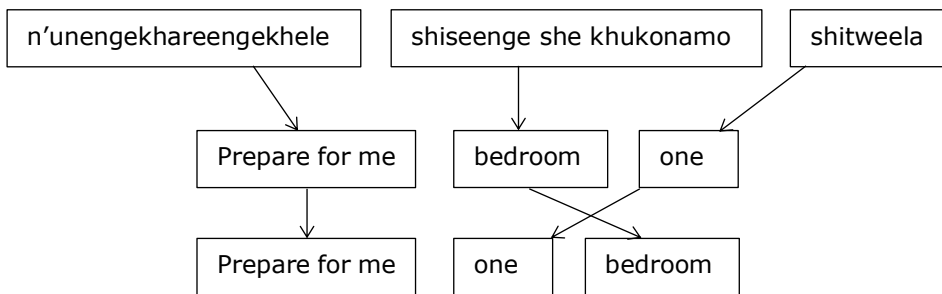


Fig. 1. Illustration of Phrase-based translation of a Lumasaaba sentence to an English sentence.

Equation 2 is representative of the breakup of the Lumasaaba sentence into n phrases and the translation of each i^{th} Lumasaaba phrase (l_i) to an English phrase (e_i) according to a phrase translation probability $\phi(l_i|e_i)$ which in terms of the noisy channel model is modeled as a translation from English-to-Lumasaaba. $\phi(l_i|e_i)$ is obtained from a phrase translation model which requires a number of steps to achieve [14]: the creation of a word alignment between each sentence pair of the parallel corpus; the extraction of phrase pairs that are consistent with the word alignment; and the

estimation of phrase translation probabilities. Since the methods that automate each of these steps are language independent, all that is required is an appropriate parallel corpus between Lumasaaba and English. Equation 2 also involves a reordering component $d(\text{start}_i - \text{end}_{i-1} - 1)$ which is obtained from a distance re-ordering model [14]. In this model, reordering is considered relative to the previous phrase. start_i denotes the position of the first word of a Lumasaaba input phrase that translates to the i^{th} English phrase and end_i is the position of the last word of the Lumasaaba phrase. As a result, the re-ordering distance is the number of words skipped (either forward or backward) when taking Lumasaaba words out of sequence. Although the probability of d can be learned from data, a data-independent exponentially decaying cost function is usually used in the case of phrase-based modeling.

The PBSMT approach has already been evaluated in translating between English and Lumasaaba [15] and for two other under-resourced Ugandan languages; Luganda [16] and Acholi [17] and others (for example in (de Pauw et al., [18]; Pa et al., [19]). The Lumasaaba-to-English translation results in [15] showed the PBSMT approach outperform other state-of-the-art methods using a limited Bible-based parallel corpus in an experimental Lumasaaba-to-English translation task. Although the PBSMT approach is relatively automated, a lot of effort is involved in combining different components to improve translation quality.

3.2.3 Neural Machine Translation

Neural Machine translation is the current state-of-the-art MT approach. Recent automatic evaluations (using the popular BLEU metric), for example from the MT shared tasks since 2016 ([20],[21],[22],[23]) shows that Neural MT significantly outperforms other MT approaches on many language pairs. Initially Artificial Neural Network approaches were hindered by limitations in computational resources. Advances in processing power in terms of Graphical Processing Units (GPUs) and increased computer storage capacity have made the ANN-based methods feasible for several applications including MT. Neural MT is now preferred and is implemented in commercial and publicly accessible Web-based MT systems. Just like other data-driven MT approaches, Neural MT requires sufficient amounts of parallel text to learn effective translation models.

At first, the most popular NMT architecture used a Recurrent Neural Network (RNN) – based encoder decoder model with attention. The basic details of this model are well described by Neubig [24]. For Lumasaaba-to-English translation, a source encoder maps each Lumasaaba word to a word vector and the word vectors can be processed to a sequence of hidden vectors. Then a decoder (for the target language) combines an RNN hidden representation of previously generated English words with the source hidden vectors to predict scores for each possible next English word. A softmax layer can then be used to produce the next English word distribution. Figure 2 illustrates this architecture in translating a Lumasaaba sentence “*renghekha shiseenge*” to an English sentence “*prepare a room*”. The mathematical formulation for this process is as follows [24]:

$$\begin{aligned}
 m_t^{(l)} = M_{.,t}^{(l)} \rightarrow h_t^{(l)} = \begin{cases} RNN^{(l)}(m_t^{(l)}, h_{t-1}^{(l)}) & t \geq 1 \\ 0 & \text{otherwise} \end{cases} \rightarrow m_t^{(e)} = M_{.,e_t-1}^{(e)} \rightarrow \\
 h_t^{(e)} = \begin{cases} RNN^{(e)}(m_t^{(e)}, h_{t-1}^{(e)}) & t \geq 1 \\ h_{|L|}^{(l)} & \text{otherwise} \end{cases} \rightarrow p_t^{(e)} = \text{softmax}(W_{h_s} h_t^{(e)} + b_s) \quad (3)
 \end{aligned}$$

The first step in Equation (3) corresponds to the mapping of Lumasaaba words to word vectors and the second step refers to the calculation of the hidden state $h_t^{(l)}$ for the i^{th} Lumasaaba word in the Lumasaaba sentence L. At the end of L, we have $h_{|L|}^{(l)}$ which is used to initialize the target RNN. The first and second step refer to the encoder phase which can be implemented in different ways including: linearly from left to right (which is the case in Equation (3)); in reverse from right to left; and bidirectionally, where we use two encoders (one traveling forward and one backward) [24]. The decoder phase starts from the third step where we look up the mapping for the target English word $m_t^{(e)}$ by also using information about the previous target English word e_{t-1} . In the fourth step, the decoder is run to calculate $h_t^{(e)}$. In the fifth step, we calculate the probability $p_t^{(e)}$ using a softmax function. Attention can be incorporated via a context vector c_t which is concatenated with $h_t^{(e)}$ to calculate the softmax distribution over the next target English words.

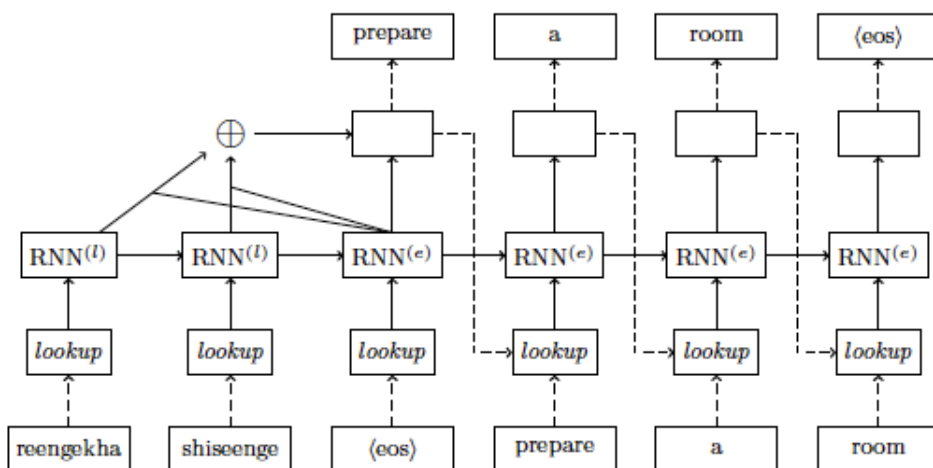


Fig. 2. An illustration of a RNN encoder-decoder architecture for translation from Lumasaaba-to-English.

The words in the Lumasaaba sentence are mapped to a word vector which is passed to a RNN. The final time step (at <eos> on the encoder side) initializes the RNN on the target side. Attention is used over the RNN at the encoder side at each time step and is combined with the current hidden state to produce the prediction of the next English word $p(e_t|e_{1:t-1}, l)$. The prediction is then fed back into the target RNN. Adapted from [25].

The RNN-based encoder-decoder model in Figure 2 has now been outperformed by the transformer model [26] in various sequence-to-sequence tasks including machine translation ([22],[23]). The transformer model which still uses an encoder-decoder approach emphasizes more attention instead of the RNN layers. Figure 3. Illustrates a transformer architecture with one encoder and decoder. The original model by Vaswani et al. [26] constitutes six encoders and six decoders. As shown in Figure 3, each layer has a multihead attention component and a simple position wise fully connected feed-forward network. The feed-forward network composes two linear transformations with either a sigmoid or ReLU activation function. The decoder side additionally has another multihead attention component that performs attention over the output from the encoder side. Embeddings in the transformer architecture are used to convert tokens (words) in the input sentence to vectors. Positional encodings (that are calculated from sine and cosine functions) are added to the input embeddings to incorporate information about relative and absolute positions of the words in

an input sentence. The “Add and normalize” layer facilitates the addition of the input to a sublayer and the normalized output from the sublayer for the final output from each sublayer. Lastly the “Linear + softmax” layer facilitates the conversion of the decoder’s output to predicted next target language words.

The transformer approach has not yet been evaluated for translation involving Lumasaaba or any other Ugandan indigenous languages. In this paper, we evaluate its performance against the RNN-based encoder decoder models that use Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) gating mechanisms.

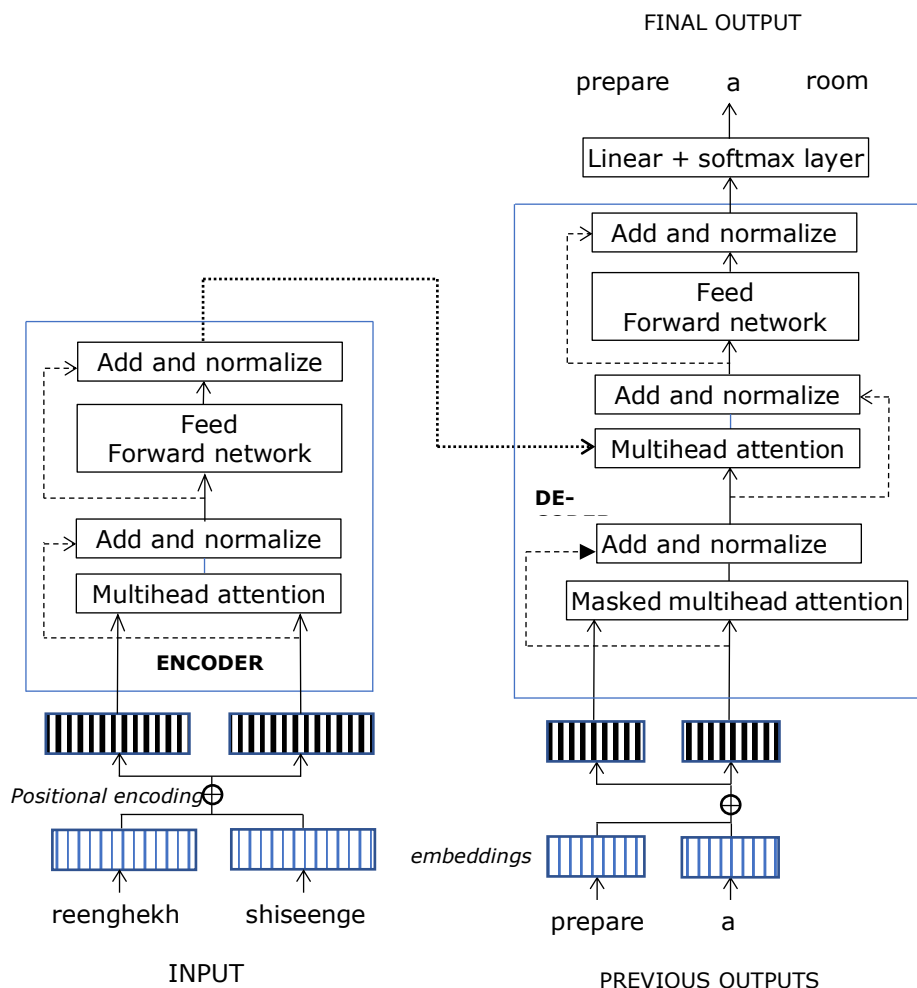


Fig. 3. An illustration of the transformer model with one encoder and decoder.

4. Data and Methodology for evaluating Neural MT methods

4.1. Data

We use an electronically accessible New Testament version of the Lumasaaba Bible to process parallel sentences for evaluating the Neural MT methods described in the previous section. Initially, we mapped verses in the Lumasaaba version of the Bible to those in the English New International Version (NIV) from which we tried to

extract parallel sentences using the full stop character (.) as a sentence separator. However, there are a number of issues to consider when using Bible versions as a source for parallel sentences including the occurrence of imperfect translations. After running a simple algorithm for extracting parallel sentences from the corresponding verses in the Lumasaaba and English NIV versions of the Bible, manual verification was done on a sample of six hundred sentences. At the start of the verification process, we observed that some pairs were not perfect translations. To deal with the imperfect cases, we sought representations from other versions of the English Bible in the hope of finding better translations. At this stage, we found that the English Good News Bible version contained more satisfactory translations for the Lumasaaba sentences than the other English Bible versions. In the end, we decided to add translations from other Bible versions, and if there was more agreement across the different versions, we decided to modify an imperfect translation to only one suitable translation. The following is an example of one such initial translation pair from the extraction script that we decided to modify:

Lumasaaba: *baryo be mu lukhabo ni baamala khuriisariisa barume musaambo tsitsindi, babalekhuula baatsya, nga sibabona isaambo ikiindi iye khubawa shitaambiso ta, lwekhuba baarya babandu*
English mapping from an extraction algorithm: *after further threats, they let them go*

As seen in the example above, the English representation from the extraction algorithm does not contain translations for other parts of the Lumasaaba sentence. In other words, it is an incomplete representation. For such a case, a consultation with other English versions of the Bible resulted in the following as an appropriate English translation for the Lumasaaba sentence:

Appropriate English translation from other versions: *so when the council had further threatened them, they let them go, as they could not find any other basis on which to punish them, because they feared the people*

On average, it took around five minutes for one person to verify and modify each translation pair or add more translations. In total it took eight days to verify and edit the sample of 600 translations (using approximately seven hours per day).

There were other cases where the differences in representations across a pair were because of other symbols. For example, there were several cases where use of braces on the source side was not the same as on the target side, or where a semi colon (;) was replaced by a comma (,) on the other side. For these cases, searches for symbols other than alphabetical characters were made and the sentences containing those symbols were checked and edited so that they are consistent across the source and target language sides.

In total, 7290 Lumasaaba-English parallel sentences resulted from the data processing described above. The small size of this data set makes it unrealistic for developing a full scale data-driven MT system. However, it is acceptable to use it in an experimental setting to evaluate MT methods. This data set (or corpus) can also be used for creating other important Natural Language Processing resources including [27]: as a source for extracting word and phrase pairs for creating Lumasaaba-English bilingual dictionaries; Named Entity Recognition between Lumasaaba and English; discourse analysis; word alignment; multi-document classification; bilingual lexical and semantic resources such as ontologies, sentiment analysis, bilingual document classification, typological or comparative language analysis; etc.

4.2. Experimental setup

The 7290 sentence pairs data set described in the previous subsection was used for evaluating the different Neural MT models described in section 3.2.3. The data set was preprocessed further and divided into three sets: a training set of 5448 parallel sentences (~75%); a validation set of 1200 parallel sentences (~16%), and a testing set of 642 parallel sentences (~9%). Further preprocessing involved tokenization (where spaces were inserted between words and other symbols), truecasing (where the sentences were all converted into the most probable casing to reduce on sparsity. Since the number of parallel sentences is small, all sentences were converted to lowercase on both sides. All NMT methods were evaluated on the same training, testing, and validation datasets. Usually, a validation phase involves further tuning of trained models to improve model performance on the validation data set; in this study, the validation dataset was used to only determine the performance of the trained models without tuning the trained models for improvement in performance.

The freely available OpenNMT-Py implementation [25] of the NMT methods was used for specifying the different architectures, and for using the trained models to generate translations. The first set of models was based on a default configuration of a unidirectional RNN encoder decoder architecture that uses two layers of 500 units each on the encoder and decoder with LSTMs as the gating mechanism. The second set of models was still based on a unidirectional RNN encoder decoder architecture but with a change in the gating mechanism from LSTMs to GRUs. In the third set of models, we used a bi-directional RNN encoder decoder architecture with LSTM units. The configurations described so far had been tested in [15] but for only models that had been generated after thirteen epochs. In this paper, we evaluate models generated from as far as 15000 training steps which is 1000 times further. Table 1 shows the standard settings for the RNN-based architecture (Figure 2). The final set of models are based on the transformer architecture (Figure 3) where we varied the number of encoder layers from one to six and the number of heads for multi-head attention between 8 and 16. Table 2 shows the standard settings for the transformer-based architecture.

All models were trained using freely accessible Google Colab GPUs. We used the adam optimization technique with residual drop out and label smoothing for all models except the default configuration from OpenNMT-Py's Quick Start demonstration which uses stochastic gradient descent (SGD). Vaswani et al. [26] observed that label smoothing makes the models more unsure as the training proceeds but at the same time improve accuracy and BLEU scores. This is indeed the case in the experiments that were conducted in this study.

Table 1. Hyper parameter settings for the RNN-based encoder decoder architectures

Parameter	Value
Word embedding size	512
Encoder type	RNN/Bidirectional RNN
Decoder type	RNN
RNN type	LSTM/GRU
Number of layers	2/4
Decoder attention	Global
Batch size	4096
Optimization techniques	adam/SGD
Learning rate	2
Learning rate decay method	noam
Dropout	0.1
Label smoothing	0.1
Maximum number of training steps (epochs)	15000

Table 2. Hyperparameter settings for the Transformer architecture

Parameter	Value
Encoder type	transformer
Decoder type	transformer
Number of encoder/decoder units	2/3/4/6
Positional encoding (using sine function)	TRUE
Adam beta 2	0.998
Number of heads for multihead attention	8/16
Size of hidden feed forward network	2042

The settings for word embedding size, optimization technique, learning rate, learning rate decay method, drop out, label smoothing, and maximum number of training steps are as indicated in Table 1.

For decoding or translation, a beam search algorithm was used without any changes to the default settings. We considered only the best translation estimated by the algorithm per input sentence.

5. Lumasaaba-to-English Neural Machine Translation Results

5.1. Training results

Figure 4 shows the progression of accuracy on training data taken at each 1000 training steps until 15000 steps. The training process for most of the models proceeds gradually with the RNN_LSTM_2L_SGD model having a smoother progression and the RNN_GRU_4L_adam model having the most relatively rougher progression. Also, the RNN_LSTM_2L_SGD model seems to peak at a considerably lower accuracy than the other models that use the adam optimization method. The transformer-based models all reach high accuracies faster than the RNN-based methods. The same can be said of Perplexity (Figure 5) where the transformer-based models have lower perplexity values than the RNN-based models in the early stages of the training process. The RNN_LSTM_2L_SGD in this case has the highest perplexity values at the start. Generally, the transformer-based models have lower perplexity values than the RNN-based methods. All models seem to converge to the same lowest perplexity value at higher training steps (10000 to 15000).

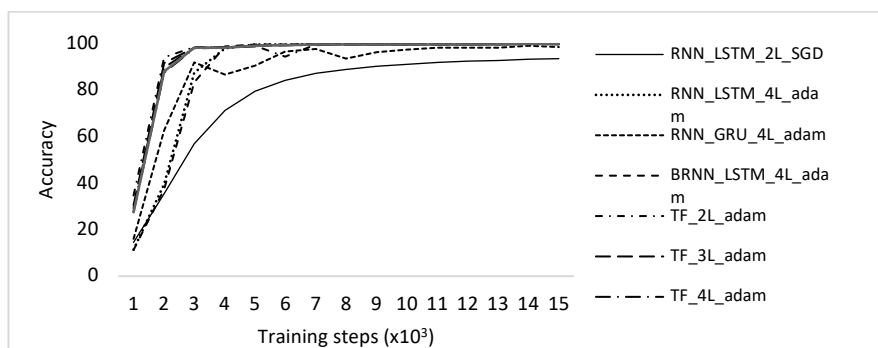


Figure 4. Progression of accuracy on training data after N training steps.

RNN_LSTM_2L_SGD refers to the default model that uses two layers of 500 units each with LSTM units and SGD for optimization. RNN_LSTM_4L_adam uses four layers of 512 units each with LSTM units and adam optimization method. RNN_GRU_4L_adam uses four layers of 512 units each with GRU units and adam optimization method. BRNN_LSTM_4L_adam uses a four layer bidirectional RNN on the encoder each with 500 units, LSTM units and adam optimization. TF_2L_adam refers to the transformer model with two encoders, two decoders, eight heads and uses adam optimization. The next transformer models TF_3L_adam to TF_6L_adam have more encoders and decoders (so that we have three encoders and three decoders for TF_3L_adam

and so on). The last model *TF_6L_16H_adam* has 16 heads with six encoders and six decoders and uses adam optimization.

5.2. Validation set results

Table 3 shows the perplexity and accuracy measures of the trained models on the validation data set after every 5000 steps of the training process. The transformer models generally have lower perplexities than the RNN-based models. As has already been mentioned, the perplexities of the models on the validation data set are bound to increase for better accuracies. Apart from the *RNN_LSTM_2L_SGD* model, we see just slight improvements in the validation accuracies which is in line with the observations of the accuracies in Figure 4 where most models have almost peaked by the 5000th training step. Although the *RNN_LSTM_2L_SGD* model has the lowest accuracies on the training data set, it achieves the highest accuracy on the validation data set.

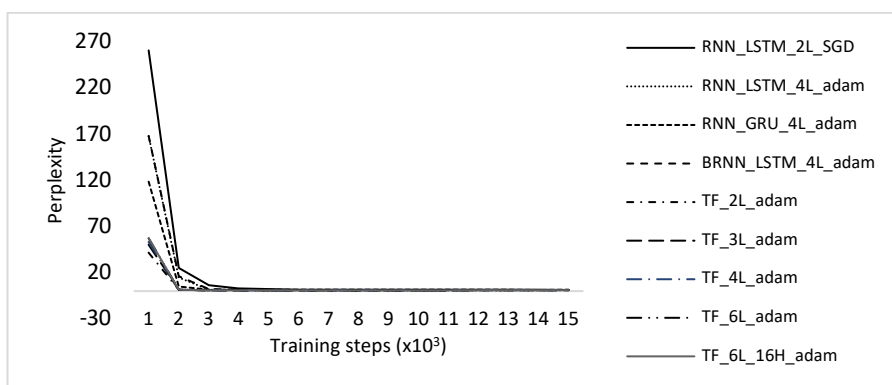


Fig. 5. Progression of perplexity on training data after N training steps.

The description for the models in this Figure (*RNN_LSTM_2L_SGD*, *RNN_LSTM_4L_adam*, *RNN_GRU_4L_adam*, *BRNN_LSTM_4L_adam*, *TF_2L_adam*, *TF_3L_adam*, *TF_4L_adam*, *TF_6L_adam*, *TF_6L_16H_adam*) is the same as in figure 4.

Table 3. Validation (Perplexity/Accuracy) results from the validation data set

Models	Perplexity/Accuracy on validation set after n training steps		
	5000	10000	15000
<i>RNN_LSTM_2L_SGD</i>	137.88/36.76	263.46/38.24	379.19/ 38.58
<i>RNN_LSTM_2L_adam</i>	387.26/26.01	425.39/27.54	466.75/27.61
<i>BRNN_LSTM_4L_adam</i>	345.22/27.27	354.56/28.75	382.39/28.70
<i>RNN_GRU_4L_adam</i>	230.14/29.90	256.56/30.73	234.61/31.07
<i>TF_2L_adam</i>	166.91/36.65	196.78/37.10	230.76/35.86
<i>TF_3L_adam</i>		201.60/36.48	218.34/36.98
<i>TF_4L_adam</i>	189.70/34.54	198.56/35.73	217.03/35.91
<i>TF_6L_adam</i>	183.29/35.03	222.57/35.21	232.73/35.54
<i>TF_6L_16H_adam</i>	188.89/34.37	224.82/34.74	258.33/34.87

The notations for the models are as specified in the description of Figure 4. Perplexity and Accuracy measures were computed after every 5000 steps of the training process. The blank cell in the Table simply means that no perplexity/accuracy was displayed for that model and hence nothing to capture at that time.

5.3. Test set results

The BLEU metric [28] which considers n-gram matches between automatically generated translations and reference translations was used for evaluating the NMT

models. The BLEU metric is defined as:

$$\text{BLEU} = \text{brevity-penalty} \times \exp\left(\sum_{i=1}^r \lambda_i \times \log(\text{precision}_i)\right)$$

$$\text{where brevity-penalty} = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases}$$

precision_i refers to the ratio of correct n-grams of a certain order n in relation to the total number of n-grams. λ_i is a weight of a specific precision, c is the length (or number of words) of an automatically generated translation, and r is the length of the reference translation. In our evaluation, λ_i is 1 for all precisions while the maximum order of n-grams to be matched is 4.

Table 4 shows the BLEU score evaluation results for the different NMT models. We again see that there are minimal variations in BLEU scores for each model after 5000 training steps. The transformer models lead to significantly higher BLEU scores than the RNN-based models that were trained using the same optimization method (adam). The highest score of 13.53 points is from a relatively simpler transformer model that uses only two encoders and two decoders. The results show that there are no performance gains from using more layers of encoders and decoders beyond the simple architecture of two encoders and decoders. The TF_6L_16H_adam model which uses more heads for attention has the lowest BLEU score among the transformer-based models; this seems to suggest that there are no MT performance gains from using more heads for attention beyond the eight heads used in the other transformer-based models. The RNN_LSTM_2L_SGD leads to comparable BLEU scores as the other transformer-based models. However, the results suggest that the transformer-based models guarantee stably better MT BLEU scores than the RNN-based models. We then evaluated the RNN_LSTM_2L_SGD and TF_2L_adam models at much higher training steps up to 100000 steps. The TF_2L_adam model posted better BLEU scores than the RNN_LSTM_2L_SGD model at lower training steps up to the 35000th training step; afterwards, the RNN_LSTM_2L_SGD model continued to improve and perform better than TF_2L_adam model at much higher training steps albeit with the highest BLEU score of 14.08 at the 80000th training step.

Table 4. Lumasaaba-to-English NMT BLEU scores using the test data set

Models	BLEU score on test set after <i>n</i> training steps		
	5000	10000	15000
RNN LSTM 2L SGD	10.30	11.75	12.74
RNN LSTM 4L adam	6.20	6.45	6.69
BRNN LSTM 4L adam	6.78	7.53	7.35
RNN GRU 4L adam	8.41	8.75	9.03
TF 2L adam	13.53	13.36	13.16
TF 3L adam	11.94	11.98	12.12
TF 4L adam	11.19	11.70	11.56
TF 6L adam	11.56	11.75	11.58
TF 6L 16H adam	10.60	10.59	10.44

The notations for the models are as specified in the description of Fig. 4.

6. Conclusion and future work

This paper set out to evaluate MT methods for automatic translation involving Lumasaaba, a heavily under-resourced East African language. Rule-based and data-driven MT methods were reviewed and opportunities and requirements for using the methods for automatic translation with Lumasaaba were identified. Neural Machine Translation (which is a state of the art machine translation method) was then used to

learn two sets of models (Recurrent Neural Network based models and Transformer based models) using a small set of parallel sentences from the Bible. Automatic evaluation of the NMT models on an experimental Lumasaaba-to-English translation task show that the transformer encoder decoder architecture leads to NMT models that guarantee stably better BLEU scores than the other NMT models at lower numbers of training steps. The work presented in this paper is the first attempt at applying currently powerful MT methods to automate translation involving Lumasaaba. The main limitation of this work is the scarcity of linguistic resources and the considerably small size of available parallel corpora for Lumasaaba which are all needed for improving MT quality. The other limitation is associated with the lack of access to more powerful GPUs for experimentation with more complicated architectures that may incorporate more helpful features. There are a number of possibilities for improving NMT that we did not explore in this paper. In future, we shall first of all continue to develop and increase the parallel sentence data set to a size that can guarantee the generation of desirable translations from data-driven MT methods. We shall also investigate the use of post-processing techniques such as the use of language models and the use of feedback for improving the quality of the automatically generated translations.

Acknowledgments

This work was supported by a Google faculty research award of 2012. I am also grateful for access to GPUs provided by Google's Collaboratory platform.

References

1. Brown, G. (1972) *Phonological Rules and Dialect Variation: A Study of the Phonology of Lumasaaba*. Cambridge University Press.
2. Purvis, J.B. (1907) *A manual of Lumasaaba Grammar*. William Clowes and Sons Limited. URL: <https://archive.org/details/AManualOfLumasabaGrammar/page/n7/mode/2up>
3. Hutchins, W. J.: *Machine Translation: History of Research and Applications*, Routledge, 2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN, chap 6, pp. 120–136. URL: www.hutchinsweb.me.uk/Routledge-2014.pdf
4. Diesner, J. (2006) Part of Speech Tagging for English text data. *Machine Learning Project Reports* pp. 1–8. URL: <https://course.ccs.neu.edu/cs4100sp12/resources/POStagging.pdf>
5. Jurafsky, D. and Martin, J.H. (2014) *Speech and Language Processing*, vol. 3. Pearson London.
6. Taylor, A., Marcus, M., Santorini, B. (2003) *The Penn TreeBank: An Overview*. Springer Netherlands, Dordrecht, pp. 5–12. URL: https://link.springer.com/chapter/10.1007%2F978-94-010-0201-1_1
7. Sleator, D.D. and Temperley, D. (1993) Parsing English with a Link Grammar. In *Proceedings of the 3rd International Workshop on Parsing Technologies*, pp. 277–292. URL: <https://www.aclweb.org/anthology/1993.iwpt-1.22.pdf>
8. Forcada, M.L., Bonev, B.I., Rojas, S.O., Ortiz, J.A.P., Sanchez, G.R., Martinez, F.S., Armentano-Oller, C., Montava, M.A., Tyers, F.M. (2010) Documentation of the Open-Source shallow-transfer machine translation platform *Apertium*. URL: <http://xixona.dlsi.ua.es/~fran/apertium2-documentation.pdf>
9. Ranta, A. (2011) *Grammatical Framework: Programming with multilingual grammars*. CLSI Publications, Stanford, California.
10. Gupta, S. (2012) A survey of data-driven machine translation. URL: www.cfilt.iitb.ac.in/resources/surveys/MT-Literature%20Survey-2012-Somya.pdf
11. Nagao, M. (1984) A framework of a mechanical translation between Japanese and English by analogy principle. *Artificial and Human Intelligence*, pp. 351–354.
12. Somers H. (1999) Review Article: Example-based Machine Translation. *Machine Translation* 14 (2): 113–157.
13. Brown, P.F., Cocke, J., Pietra, S.A., Pietra, V.J.D., Jelinek F., Lafferty, J.D., Mercer, R.L., Roosin, P.S. (1990) A Statistical Approach to Machine Translation. *Comput Linguist* 16 (2): 79–85. URL: <https://www.aclweb.org/anthology/J90-2002.pdf>
14. Koehn, P. (2010) *Statistical Machine Translation*. Cambridge University Press, UK.

15. Nabende, P. (2019) Towards data-driven machine translation for Lumasaaba. In: Antipova, T. and Rocha, A. (editors) *Digital Science*. DSIC18 2018. Advances in Intelligent Systems and Computing, vol. 850, pp. 3–11. Springer, Cham. URL: https://link.springer.com/chapter/10.1007/978-3-030-02351-5_1
16. Katende, J. (2015) Phrase-based Machine Translation between Luganda and English. Masters thesis, Makerere University, Kampala, Uganda.
17. Akello, C.K. (2017) Computational models for phrase-based statistical machine translation between Acholi and English. Masters thesis, Makerere University, Kampala, Uganda.
18. de Pauw, G., Maajabu, N., Wagacha, P.W. (2010) A knowledge-light approach to Luo Machine Translation and Part-of-Speech tagging. In *Proceedings of the second workshop on African Language Technology*, European Language Resources Association, Valletta, Malta, pp. 15–20.
19. Pa, W.P., Thu, Y.K., Finch, A., Sumita, E. (2016) A study of statistical machine translation methods for under-resourced languages. *Procedia Computer Science* 81:250–257, SLTU-2016 5th Workshop on spoken language technologies for under-resourced languages, Yogyakarta, Indonesia. URL: <https://www.sciencedirect.com/science/article/pii/S1877050916300710>
20. Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A.J., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., Zampieri, M. (2016) Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, Volume 2: Shared Task Papers, pp. 131–198, Berlin, Germany. Association for Computational Linguistics. URL: <https://www.aclweb.org/anthology/W16-2301.pdf>
21. Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., Turchi, M. (2017) Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the second Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pp. 169–214, Copenhagen, Denmark. Association for Computational Linguistics. <https://www.aclweb.org/anthology/W17-4717.pdf>
22. Bojar, O., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Monz, C. (2018) Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pp. 272--303, Belgium Brussels. Association for Computational Linguistics. URL: <https://www.aclweb.org/anthology/W18-6401.pdf>
23. Barrault, L., Bojar, O., Costa-jussà, M.R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., Zampieri, M. (2019) Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the fourth Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers (Day 1), pp. 1–61, Florence, Italy. Association for Computational Linguistics. URL: <https://aclweb.org/anthology/W19-5301.pdf>
24. Neubig, G. (2017) Neural Machine Translation and Sequence-to-Sequence models: A tutorial. CoRR abs/1703.01619, URL: <http://arxiv.org/abs/1703.01619>
25. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.M. (2017) OpenNMT: Open-source toolkit for Neural Machine Translation. ArXiv e-prints 1701.02810.
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). *Attention is all you need*. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., editors, *Advances in Neural Information Processing Systems* 30, pp. 5998–6008. Curran Associates, Inc.
27. Christodouloupoulos and Steedman, M. (2015) A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, 49 (2): 375–395.
28. 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 on Association for Computational Linguistics*, pp. 311–318, Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02.

Aims and Objectives

Published online by ICS two times a year, Journal of Digital Science (JDS) is an international peer-reviewed journal which aims at the latest ideas, innovations, trends, experiences and concerns in the field of digital science covering all areas of the scholarly literature of the sciences, social sciences. The main topics currently covered include: Digital Communications and Network; Digital Economics, Education, Engineering, Finance, Health Care.

The main goal of the journal is the effective dissemination of original incites/results generated by the human brain and presented/reflected in articles using modern information/digital technology.

Editorial Board

Editor-in-Chief Tatiana Antipova, ICS,
<https://orcid.org/0000-0002-0872-4965>

Associate Editor Julia Belyasova, Catholic University of Louvain, Louvain-la-Neuve, Belgium;
<https://orcid.org/0000-0001-6983-2129>

Editors

Abdulsatar Sultan, Catholic University in Erbil, Erbil, Iraq;
<https://orcid.org/0000-0001-5090-5332>

Jelena Jovanovic, University of Nis, Nis, Serbia;
<https://orcid.org/0000-0001-7238-6393>

Lucas Tomczyk, Uniwersytet Jagielloński, Krakow, Poland
<https://orcid.org/0000-0002-5652-1433>

Natalya Sukurova, State University of Telecommunications, Kyiv, Ukraine
<https://orcid.org/0000-0003-4297-1123>

Olga Khlynova, Russian Academy of Science, Moscow, Russia
<https://orcid.org/0000-0003-4860-0112>

Omar Leonel Loaiza Jara, Universidad Peruana Unión, Lima, Peru
<https://orcid.org/0000-0002-3262-709X>

Roland Moraru, University of Petrosani, Romania
<https://orcid.org/0000-0001-8629-8394>

Tjerk Budding, Vrije Universiteit Amsterdam, Netherland
<https://orcid.org/0000-0002-5343-7535>

Zhanna Mingaleva, National Research Polytechnic University, Perm, Russia
<https://orcid.org/0000-0001-7674-7846>

Quang Vinh Dang, Industrial University, Ho Chi Minh City, Viet Nam
<https://orcid.org/0000-0002-3877-8024>

Contact information

Website: <https://ics.events>

Email: conf@ics.events