# Low-resource neural machine translation with morphological modeling

**Antoine Nzeyimana**
University of Massachusetts Amherst
anthonzeyi@gmail.com

## Abstract

Morphological modeling in neural machine translation (NMT) is a promising approach to achieving open-vocabulary machine translation for morphologically-rich languages. However, existing methods such as sub-word tokenization and character-based models are limited to the surface forms of the words. In this work, we propose a framework-solution for modeling complex morphology in low-resource settings. A two-tier transformer architecture is chosen to encode morphological information at the inputs. At the target-side output, a multi-task multi-label training scheme coupled with a beam search-based decoder are found to improve machine translation performance. An attention augmentation scheme to the transformer model is proposed in a generic form to allow integration of pre-trained language models and also facilitate modeling of word order relationships between the source and target languages. Several data augmentation techniques are evaluated and shown to increase translation performance in low-resource settings. We evaluate our proposed solution on Kinyarwanda ↔ English translation using public-domain parallel text. Our final models achieve competitive performance in relation to large multi-lingual models. We hope that our results will motivate more use of explicit morphological information and the proposed model and data augmentations in low-resource NMT.

## 1 Introduction

Neural Machine Translation (NMT) has become a predominant approach in developing machine translation systems. Two important innovations in recent state-of-the-art NMT systems are the use of the Transformer architecture (Vaswani et al., 2017) and sub-word tokenization methods such as byte-pair encoding (BPE) (Sennrich et al., 2016). However, for morphologically-rich languages(MRLs), BPE-based tokenization is only limited to the surface forms of the words and less grounded on exact lexical units (i.e. morphemes), especially in the presence morphographemic alternations (Bundy and Wallen, 1984) and non-concatenative morphology (Kastner et al., 2019). In this work, we tackle the challenge of modeling complex morphology in low-resource NMT and evaluate on Kinyarwanda, a low-resource and morphologically-rich language spoken by more than 15 million people in Eastern and Central Africa[1].

To model the complex morphology of MRLs in machine translation, one has to consider both source-side modeling (i.e. morphological encoding) and target-side generation of inflected forms (i.e. morphological prediction). We explicitly use the morphological structure of the words and the associated morphemes, which form the basic lexical units. For source-side encoding, morphemes are first produced by a morphological analyzer before being passed to the source encoder through an embedding mechanism. On the target side, the morphological structure must be predicted along with morphemes, which are then consumed by an inflected form synthesizer to produce surface forms. Therefore, this approach enables open vocabulary machine translation since morphemes can be meaningfully combined to form new inflected forms not seen during training.

Previous research has shown that certain adaptations to NMT models, such as the integration of pre-trained language models (Zhu et al., 2020; Sun et al., 2021), can improve machine translation performance. We explore this idea to improve low-resource machine translation between Kinyarwanda and English. Our model augmentation focuses on biasing the attention computation in the transformer model. Beside augmentation from pre-trained language model integration, we also devise an augmentation based solely on the

[1]https://en.wikipedia.org/wiki/Kinyarwanda



*Findings of the Association for Computational Linguistics: NAACL 2024*, pages 182–195
June 16-21, 2024 ©2024 Association for Computational Linguistics

word order relationship between source and target languages. These model augmentations bring substantial improvement in translation performance when parallel text is scarce.

One of the main challenges facing machine translation for low-resource languages obviously is parallel data scarcity. When the training data has limited lexical coverage, the NMT model may tend to hallucinate (Raunak et al., 2021; Xu et al., 2023). Additionally, for a morphology-aware translation model, there is a problem of misaligned vocabularies between source and target languages. This makes it harder for the model to learn to copy unknown words and other tokens that need to be copied without translation such as proper names. To address these challenges, we take a data-centric approach by developing tools to extract more parallel data from public-domain documents and websites. We also use various data augmentation techniques to increase lexical coverage and improve token copying ability where necessary. By combining these data-centric approaches with our morphology-aware NMT model, we achieve competitive translation performance in relation to larger multi-lingual NMT models. To have a comprehensive evaluation, we evaluate our models on three different benchmarks covering different domains, namely Wikipedia, News and Covid-19.

In short, our contribution in this work can be summarized as follow:

- We propose and evaluate methods for source-side and target-side morphological modeling in neural machine translation of morphologically-rich languages.
- We propose a generic method for attention augmentation in the transformer architecture, including a new cross-positional encoding technique to fit word order relationships between source and target language.
- We evaluate on Kinyarwanda↔English translation across three benchmarks and achieve competitive performance in relation to existing large multi-lingual NMT models.
- We release tools for parallel corpus construction from public-domain sources and make our source code publicly available to allow reproducibility[2].

[2] https://github.com/anzeyimana/KinMT_NAACL2024

## 2 Methods

Machine translation (MT) can be considered as the task of accurately mapping a sequence of tokens (e.g. phrase, sentence, paragraph) in the source language $S = (s_1, s_2, ...s_n)$ to a sequence of tokens in the target language $T = (t_1, t_2, ...t_m)$ with the same meaning. The learning problem is then to estimate a conditional probability model that produces the optimal translation $T^*$, that is:

$$T^* = \arg\max_T P(T|S, T_<; \Theta),$$

where $T_<$ accounts for the previous output context and $\Theta$ are parameters of the model (that is a neural network in the case of NMT).

In this section, we describe our model architecture as an extension of the basic Transformer architecture (Vaswani et al., 2017) to enable morphological modeling and attention augmentation. We also describe our data-centric approaches to dataset development and augmentation in the context of the low-resource Kinyarwanda↔English machine translation.

### 2.1 Model architecture

The transformer architecture (Vaswani et al., 2017) for machine translation uses a multi-layer bidirectional encoder to process source language input, and then feeds to an auto-regressive decoder to produce the target language output. Our adaptation of the transformer encoder is depicted in Figure 1 while the decoder is shown in Figure 2. They both use pre-LayerNorm configuration (Nguyen and Salazar, 2019) of the transformer.

The **attention module** of the transformer architecture is designed as querying a dictionary made of key-value pairs using a softmax function and then projecting a weighted sum of value vectors to an output vector, that is:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V, \quad (1)$$

where $K$,$Q$,$V$ are projections of the hidden representations of inputs at a given layer. Given a hidden representation of a token $v_i$ attending to a sequence of tokens with hidden representations $(w_1, w_2, ...w_n)$, the output of the attention module corresponding to $v_i$ can be formulated as:

$$v_i' = \sum_{j=1}^{n} \frac{\exp(\alpha_{ij})}{\sum_{j'=1}^{n} \exp(\alpha_{ij'})}(w_j W_V),$$
$$\text{where the logits } \alpha_{ij} = \frac{1}{\sqrt{d}}(v_i W_Q)(w_j W_K)^T, \tag{2}$$

with $W_Q \in \mathbb{R}^{d \times d_K}, W_K \in \mathbb{R}^{d \times d_K}$, and $W_V \in \mathbb{R}^{d \times d_V}$ being learnable projection matrices. $d, d_K$ and $d_V$ are the dimensions of the input, key and value respectively.

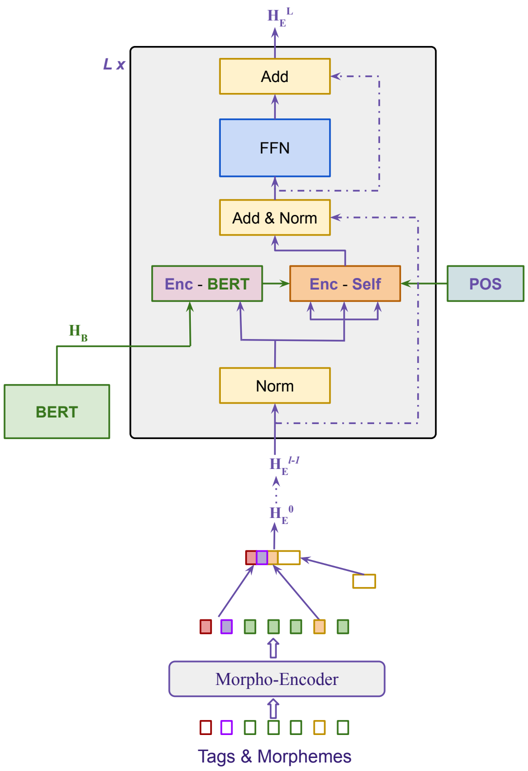


Figure 1: Encoder architecture

### 2.1.1 Attention augmentations

Ke et al. (2020) proposed to add bias terms to the logits $\alpha_{ij}$ in Equation 2 as untied positional encoding, disentangling a mixing of token and position embeddings. We generalize this structure by allowing more biases to be added to the logits $\alpha_{ij}$ in Equation 2.

Specifically, we explore augmenting two attention components in the transformer architecture by making the following extensions:

1. *For **Source-to-source self-attention** at each encoder layer*: We integrate embeddings from a pre-trained BERT (Devlin et al., 2019) model. This adds rich contextual information as BERT models are pre-trained on large monolingual data and perform well on language understanding tasks. We also add positional encodings at this level, similar to (Ke et al., 2020). Therefore, the logits $\alpha_{ij}$ at encoder layer $l$ become:

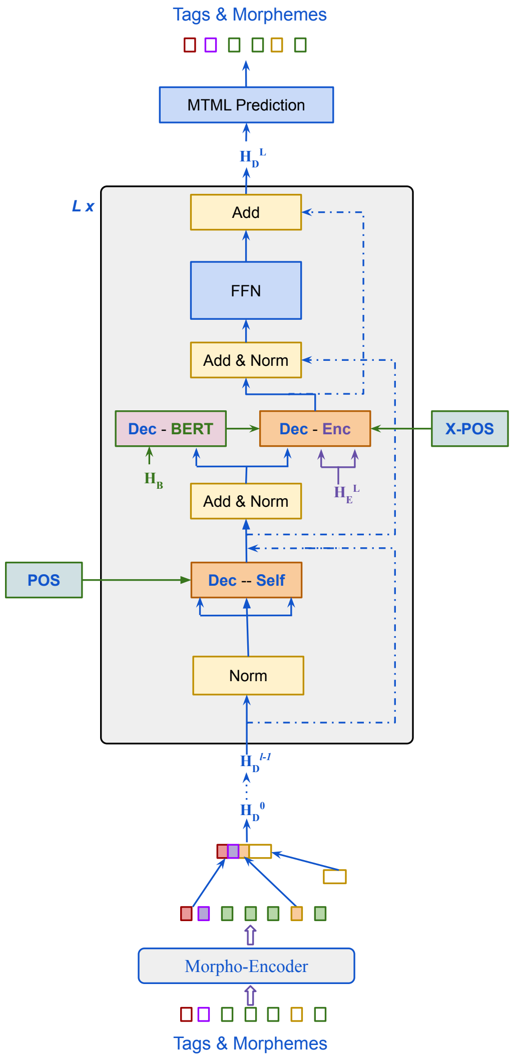


Figure 2: Decoder architecture

$$\alpha_{ij} = \frac{1}{\sqrt{3d}}[(x_i^{(l)} W_Q^{(l)})(x_j^{(l)} W_K^{(l)})^T + (p_i U_Q)(p_j U_K)^T + r_{j-i} + (x_i^{(l)} V_Q^{(l)})(b_j V_K^{(l)})^T], \tag{3}$$

where $x_i^{(l)}$ and $x_j^{(l)}$ are hidden representations of source tokens at positions $i$ and $j$ respectively of the encoder layer $l$; $p_i$ and $p_j$ are absolute position embeddings; $r_{j-i}$ is a relative position embedding; $b_j$ is a pre-trained BERT embedding of token at position $j$ and

$W_Q^{(l)}, W_K^{(l)}, U_Q, U_K, V_Q^{(l)}$ and $V_K^{(l)}$ are learnable projection matrices. We note that this formulation requires the source encoder to match the same token vocabulary as the BERT embedding model.

2. *For* ***target-to-source cross-attention*** *at each decoder layer $l$*, we also augment the attention logits with pre-trained BERT embeddings of the source sequence. Additionally, we propose a new type of embedding: **cross-positional embeddings**. These are embeddings that align target sequence positions to input sequence positions. Their role can be thought as of learning word order relationships between source and target languages. Their formulation is closely similar to the untied positional encoding proposed by (Ke et al., 2020), but they cross from target to source positions, thus, we name them cross-positional (XPOS) encodings. The attention logits $\alpha_{ij}^{'}$ at this level thus become:

$$\begin{aligned} \alpha_{ij}^{'} = \frac{1}{\sqrt{3d}}[(y_i^{(l)} W_Q^{'(l)})(x_j^{(L)} W_K^{'(l)})^T \\ + (p_i^{'} U_Q^{'})(p_j^{'} U_K^{'})^T + r_{j-i}^{'} \\ + (y_i^{(l)} V_Q^{'(l)})(b_j V_K^{'(l)})^T], \end{aligned} \tag{4}$$

where $y_i^{(l)}$ is the hidden representation of the target token at position $i$ and $x_j^{(L)}$ is the hidden representation of source token at position $j$ of the final encoder layer $L$. $p_i^{'}$ and $p_j^{'}$ are absolute target and source XPOS embeddings, $r_{j-i}^{'}$ is a target-to-source relative XPOS embedding, $b_j$ is a pre-trained BERT embedding of source token at position $j$ and $W_Q^{'(l)}, W_K^{'(l)}, U_Q^{'}, U_K^{'}, V_Q^{'(l)}$ and $V_K^{'(l)}$ are the learnable projection matrices.

### 2.1.2 Morphological encoding

For most transformer-based encoder-decoder models, the first layer inputs are usually formed by mapping each sub-word token, such as those produced by BPE, to a learnable embedding vector. However, BPE-produced tokens do not always carry explicit lexical meaning. In fact, they cannot model non-concatenative morphology and other morphographemic processes as these BPE tokens are solely based on the surface forms. Inspired by the work of Nzeyimana and Rubungo (2022), we explore using a small transformer encoder to form a word-compositional model based on the morphological structure and the associated morphemes.

Depicted at the input layers in Figure 1 and Figure 2, the morphological encoder or *Morpho-Encoder* is a small transformer encoder that processes a set of four embedding units at the word composition level: (1) **the stem**, (2) a **variable number of affixes**, (3) a coarse-grained **part-of-speech (POS) tag** and, (4) a fine-grained **affix set index**. An affix set represents one of many frequent affix combinations observed empirically. Thus, the affix set index is equivalent to a fine-grained **morphological tag**.

The *Morpho-Encoder* processes all word-compositional units as a set without any ordering information. This is because none of these units can be repeated in the same word. In cases of stem reduplication phenomena (Inkelas and Zoll, 2000), only one stem is used, while the reduplication structure is captured by the affix set. At the output of the *Morpho-Encoder*, hidden representations corresponding to units other than the affixes are pulled and concatenated together to form a word hidden vector to feed to the main sequence model. In addition to this, a new stem embedding vector at the sequence level is also concatenated with the pulled vectors from the *Morpho-Encoder* to form the final hidden vector representing the word.

In our experiments, we use 24,000 most frequent affix combinations as affix sets. Any infrequent combination of affixes can always be reduced to a frequent one by removing one or more affixes. However, all affixes still contribute to the word composition via the *Morpho-Encoder*.

We note that the *Morpho-Encoder* applies to both the encoder and the auto-regressive decoder's input layers for all types of tokens. While the word-compositional model above relates mostly to inflected forms, we are able generalize this to other typed of tokens such as proper names, numbers and punctuation marks. For these other tokens, we process them using BPE and consider the resulting sub-word tokens as special stems without affixes.

### 2.1.3 Target-side morphology learning

Considering the morphological model employed at the input layer, the decoder outputs for a morphologically-rich target language must be used to predict the same types of morphological units used at the input layer, namely, the stem, affixes, the POS tag and the affix set. This becomes a multi-task and multi-label (MTML) classification problem which requires optimizing multiple objectives, corresponding to 4 types loss functions:

$$\begin{aligned}\ell_S &= \ell_{CE}(f_S(h^L), y_S)\\ \ell_A^{(i)} &= \ell_{BCE}(f_A(h^L), y_A^{(i)}), \forall i \in \mathcal{A}\\ \ell_P &= \ell_{CE}(f_P(h^L), y_P)\\ \ell_{AS} &= \ell_{CE}(f_{AS}(h^L), y_{AS}),\end{aligned} \tag{5}$$

where $h^L$ is the decoder output, $\mathcal{A}$ is the set of affix indices, $f_S, f_A, f_P$, and $f_{AS}$ are prediction heads transforming the decoder output to probabilities over the sets of stems, affixes, POS tags and affix sets respectively. $y_S, y_A, y_P$, and $y_{AS}$ respectively correspond to the stem, affixes, POS tags and affix set of a target word $y$. $\ell_{CE}$ is a cross-entropy loss function and $\ell_{BCE}$ is a binary cross-entropy loss function.

A naive approach to the MTML problem consists of summing up all the losses and optimizing the sum. However, this can lead to a biased outcome since individual losses take on different ranges and have varying levels of optimization difficulty. Complicating the problem further is the fact that individual objectives can contribute conflicting gradients, making it harder to train the multi-task model with standard gradient descent algorithms. A potential solution to this problem comes form the multi-lingual NMT literature with a scheme called Gradient Vaccine (Wang et al., 2020). This method attempts to mediate conflicting gradient updates from individual losses by encouraging more geometrically aligned parameter updates. We evaluate both the naive summation and the Gradient Vaccine methods in our experiments.

#### 2.1.4 Morphological inference

The decoder architecture presented in subsection 2.1.3 only predicts separate probabilities for stem, POS tag, affix set and affixes. But the translation task must produce surface forms to generate the output text. The challenge of this task is that greedily picking the items with maximum probability may not produce the best output and may even produce incompatible stems and affixes, that is, we must produce a stem and affixes of the same inflection group (e.g. verb, noun, pronoun, etc..). It is also known that beam search algorithm generally produces better sequence outputs than greedy decoding. Therefore, we design an adaptation of the beam search algorithm, where at each step, we produce a list of scored candidate surface forms together with their morphological information to feed back to the decoder's input. The design criteria is to make sure the top predicted items can form compatible pairs of stems and affix sets. The main requirement for the algorithm is the availability of a morphological synthesizer that can produce surface forms given an inflection group, the stem and compatible affixes. The morphological synthesizer must also respect all existing morphographemic rules for the language. We we provide detailed pseudocode for the decoding algorithm in Appendix C. The algorithm has 4 basic steps:

1. voting on inflection group
2. filtering out less probable stems and affixes
3. selecting target affixes, and finally
4. morphological synthesis for each final stem and affixes combination.

### 2.2 Dataset

Dataset development and data-centric approaches to neural machine translation (NMT) are of paramount importance for low-resource languages. This is because the most limiting factor is the scarcity of parallel data. While describing the data collection process and pre-processing steps is important, it is equally important to fully disclose the data provenance as there are typically a limited number of sources of parallel data per language. We conduct our experiments using public-domain parallel text. In this section, we describe our parallel data gathering process as well as the reliable sources we used to source Kinyarwanda-English bitext. We also describe simple data augmentation techniques we used to boost the performance of our experimental models. Due to copyright and licensing restrictions, we cannot redistribute our experimental dataset. Instead, we release the tools used for their construction from original sources. The sizes of the parallel datasets we gathered are provided in Appendix B.

### 2.3 Official Gazette

Official gazettes are periodical government journals typically with policy and regulation content. When a country has multiple official languages, content may be available as parallel text with each paragraph of the journal available in each official language. We took this opportunity and collected an experimental parallel text from the Official Gazette of the Republic of Rwanda[3], where Kinyarwanda,

[3] https://www.minijust.gov.rw/official-gazette

French, English and Swahili are all official languages. This is an important source of parallel text given that it covers multiple sectors and is usually written with high standards by professionals, part of a dedicated government agency.

The main content of Rwanda's official gazette is provided in a multi-column portable document format (PDF), mostly 3 columns for Kinyarwanda, English and French. In our experiments, we process page content streams by making low level modifications to Apache PDFBox Java library[4], where the inputs are unordered set of raw characters with their X-Y page coordinates and font information. We track columns by detecting margins (by sorting X-coordinates of glyphs) and reconstruct text across consecutive pages. A key opportunity for parallel alignment comes from the fact that most official gazettes paragraphs are grouped by consecutive article numbers such as "Ingingo ya 1/Article 1", "Ingingo ya 2/Article 2", and so on. We use these article enumerations as anchors to finding parallel paragraphs across the three languages. A language identification component is also required to know which column correspond with which language as the column ordering has been changing over time.

### 2.4 Jw.org website

Jw.org website publishes religious and biblical teachings by Jehovah's Witnesses, with cross-references into multiple languages. While this website data has been used for low-resource machine translation before (Agić and Vulić, 2019), the isolated corpus is no longer available due to license restrictions. However, the content of the original website is still available to web browsers and crawlers. We take this advantage and gather data from the site to experiment with English↔Kinyarwanda machine translation.

### 2.5 Bilingual dictionaries

Bilingual dictionaries are also useful for low-resource machine translation. While most of their parallel data are made of single words, they can still contribute to the translation task, albeit without any sentence-level context. The 2006 version of the Iriza dictionary (Habumuremyi and Uwamahoro, 2006) is generally publicly available in PDF format. Similar to the Official Gazette case, we use low level modifications to the Apache PDFBox library and extract dictionary entries grouped by a source word and a target synset. Another bilingual dictionary we used is *kinyarwanda.net* website[5], which was developed by volunteers to help people learning Kinyarwanda or English. In addition to these bilingual dictionaries, we manually translated about 8,000 Kinyarwanda words whose stems could not be found in any of the existing parallel data sources. Some of these terms include recently incorporated but frequently used Kinyarwanda words such as loanwords and also alternate common spellings. Examples include words such as: 'abazunguzayi' (*hawkers*), 'akabyiniro' (*night club*), 'canke' (from Kirundi: *or*), 'mitiweli' (from French: "mutuelle santé"). Together with data from bilingual dictionaries, the above dataset forms a special training subset we call 'lexical data', because it augments the lexical coverage of our main dataset. We evaluate its effectiveness in our experiments.

### 2.6 Monolingual data

Backtranslation (Edunov et al., 2018) is a proven technique for leveraging monolingual data in machine translation. We developed a corpus of Kinyarwanda text by crawling more than 200 websites and extracting text from several books to form a monolingual dataset to use for back-translation. The final corpus contains about 400 million words and tokens or 16 million sentences. We also formed an English text corpus of similar size by crawling eight major Rwandan and East African English newspapers (3.3 million sentences) in addition to Wikipedia English corpus (7.3 million sentences)[6] and global English news data (5.4 million sentences)[7].

### 2.7 Data augmentations

Source to target copying in NMT is a desirable ability when faced with untranslatable terms such as proper names. However, when the source and target vocabularies are not shared, it is harder for the model to learn this ability. In order to enforce this copying ability in our NMT model, we take a data-centric approach by including untranslatable terms in our dataset with the same source and target text. This augmentation includes the following

[4] https://pdfbox.apache.org/

[5] https://kinyarwanda.digital/

[6] https://www.kaggle.com/datasets/mikeortman/wikipedia-sentences

[7] https://data.statmt.org/news-crawl/en/news.2020.en.shuffled.deduped.gz

datasets: (1) All numeric tokens and proper names from our Kinyarwanda monolingual corpus, (2) Names of locations from the World Cities dataset[8], and (3) Names of people from CMU Names corpus (Kantrowitz and Ross, 2018) and the Names Dataset (Remy, 2021).

We also add synthetic data for number spellings by using rule-based synthesizers to spell 200,000 random integers between zero and 999 billions. For Kinyarwanda side, we developed our own synthesizer, while we used `inflect` python package[9] for English.

Code-switching is one characteristic of some low-resource languages such as Kinyarwanda. To cope with this issue, we add foreign language terms and their English translations to the training dataset for Kinyarwanda → English models as if the foreign terms were valid Kinyarwanda inputs. For this, we include all English phrases from *kinyarwanda.net* online dictionary, 100 popular French terms and 20 popular Swahili terms.

## 3 Experiments

### 3.1 Experimental setup

The model presented in section 2.1 was implemented from scratch using PyTorch framework (Paszke et al., 2019) version 1.13.1. Our model hyper-parameters along with training and inference hyper-parameters are provided in appendix A. Training was done using a hardware platform with 8 Nvidia RTX 4090 GPUs, with 256 gigabytes of system memory, on a Linux operating system. We used mixed precision training with lower precision in BFLOAT-16 format. For Kinyarwanda→English model, one gradient update step takes 0.5 second and convergence is achieved after 40 epochs. For English→Kinyarwanda model with Gradient Vaccine scheme, one gradient update step takes 1.1 seconds while convergence is achieved after 8 epochs.

In all our experiments on Kinyarwanda ↔ English translation, only Kinyarwanda side (as source or target) is morphologically modelled, while the English side always uses sub-word tokenization. For Kinyarwanda source side with attention augmentation, we use a pre-trained BERT model similar to KinyaBERT (Nzeyimana and Rubungo, 2022) whose input units/token ids are the same as for the NMT encoder. In fact, this pre-trained KinyaBERT model has the same two-tier architecture as the NMT encoder. Therefore, they are aligned to the same words/tokens. We use a Kinyarwanda morphological analyzer[10] to perform both tokenization, morphological analysis and disambiguation (Nzeyimana, 2020).

On English sides (either source or target), we do not perform morphological analysis and only use a standard single-tier transformer architecture. On the source side, we use a BPE-based tokenization and a corresponding pre-trained RoBERTA model provided by fairseq package (Ott et al., 2019). Similarly, on English target side, we use a BPE-based tokenization from a Transformer-based auto-regressive English language model (Ng et al., 2019) from the same fairseq package.

### 3.2 Evaluation

We evaluate our models on three different benchmarks that include Kinyarwanda, namely FLORES-200 (Costa-jussà et al., 2022), MAFAND-MT (Adelani et al., 2022) and TICO-19 (Anastasopoulos et al., 2020). This allow us to have a picture on how the models perform on different domains, respectively Wikipedia, News and Covid-19. Our main evaluation metric is ChrF++ (Popović, 2017) which includes both character-level and word-level n-gram evaluation, does not rely to any sub-word tokenization and has been shown to correlate better with human judgements than the more traditional BLUE score. We use TorchMetrics (Detlefsen et al., 2022) package's default implementation of ChrF++. For Kinyarwanda→ English translation, we also evaluate with BLEURT scores (Sellam et al., 2020), an embedding-based metric with higher correlation with human judgement. We use a pre-trained PyTorch implementation of BLEURT[11]. We did not use BLEURT scores for English→Kinyarwanda because there was no available pre-trained BLEURT model for Kinyarwanda and the pre-training cost is very high.

The baseline BPE-based models in Table 4 and Table 5 use a SentencePiece (Kudo and Richardson, 2018) tokenizer, with 32K-token vocabularies for either source or target. The SentencePiece tokenizers are trained/optimized on 16M sentences of text for each language. Source and target vocabularies are not shared. The NMT models

[8] https://github.com/datasets/world-cities/blob/master/data/world-cities.csv

[9] https://pypi.org/project/inflect/

[10] https://github.com/anzeyimana/DeepKIN

[11] https://github.com/lucadiliello/bleurt-pytorch

| Use copy data? | FLORES-200 BLEURT | FLORES-200 ChrF++ | MAFAND-MT BLEURT | MAFAND-MT ChrF++ | TICO-19 BLEURT | TICO-19 ChrF++ | **Average** BLEURT | **Average** CHR++ |
|---|---|---|---|---|---|---|---|---|
| No | 55.5 | 39.2 | 52.3 | 37.1 | 52.2 | 33.4 | 53.3 | 36.6 |
| Yes | **56.8** | **40.3** | **54.8** | **39.6** | **52.9** | **34.0** | **54.9** | **38.0** |

Table 1: Impact of proper name copying ability: **Kinyarwanda** → **English**. Maximum scores are shown in bold.

| Setup | **#Params** (× **1M**) | FLORES-200 BLEURT | FLORES-200 ChrF++ | MAFAND-MT BLEURT | MAFAND-MT ChrF++ | TICO-19 BLEURT | TICO-19 ChrF++ | **Average** BLEURT | **Average** CHR++ |
|---|---|---|---|---|---|---|---|---|---|
| Morpho | 188 | 57.1 | 40.9 | 54.7 | 39.8 | 53.1 | 34.7 | 55.0 | 38.5 |
| + XPOS | 190 | 57.7 | 41.1 | 55.6 | 39.9 | 54.1 | 35.3 | 55.8 | 38.8 |
| + BERT | 190 | 59.4 | 42.5 | 57.1 | 40.4 | 56.1 | 36.3 | 57.5 | 39.7 |
| + BERT + XPOS | 192 | **59.9** | **43.1** | **58.0** | **41.3** | **56.7** | **37.0** | **58.2** | **40.5** |

Table 2: Impact of attention augmentation: **Kinyarwanda** → **English**

| **Setup** | FLORES-200 BLEURT | FLORES-200 ChrF++ | MAFAND-MT BLEURT | MAFAND-MT ChrF++ | TICO-19 BLEURT | TICO-19 ChrF++ | **Average** BLEURT | **Average** CHR++ |
|---|---|---|---|---|---|---|---|---|
| Morpho + BERT + XPOS without Lexical Data | 58.5 | 41.6 | 56.3 | 39.9 | 55.5 | 36.0 | 56.8 | 39.2 |
| Morpho + BERT + XPOS + Lexical Data | **59.9** | **43.1** | **58.0** | **41.3** | **56.7** | **37.0** | **58.2** | **40.5** |

Table 3: Impact of lexical data (bilingual dictionaries): **Kinyarwanda** → **English**

| **Setup** | **#Params** (× **1M**) | FLORES-200 BLEURT | FLORES-200 ChrF++ | MAFAND-MT BLEURT | MAFAND-MT ChrF++ | TICO-19 BLEURT | TICO-19 ChrF++ | **Average** BLEURT | **Average** CHR++ |
|---|---|---|---|---|---|---|---|---|---|
| BPE Seq2Seq + XPOS | 187 | 50.1 | 35.5 | 48.5 | 34.2 | 47.4 | 30.7 | 48.7 | 33.5 |
| Morpho + XPOS | 190 | **57.7** | **41.1** | **55.6** | **39.9** | **54.1** | **35.3** | **55.8** | **38.8** |

Table 4: Impact of source side morphological modeling: **Kinyarwanda** → **English**

| **Setup** | **#Params** (× **1M**) | FLORES-200 Dev ChrF++ | FLORES-200 Test ChrF++ | MAFAND-MT Dev ChrF++ | MAFAND-MT Test ChrF++ | TICO-19 Dev ChrF++ | TICO-19 Test ChrF++ | **Average** Dev ChrF++ | **Average** Test ChrF++ |
|---|---|---|---|---|---|---|---|---|---|
| BPE Seq2Seq + XPOS | 187 | 35.0 | 35.2 | 37.0 | 37.8 | 30.1 | 31.1 | 34.0 | 34.7 |
| Morpho + XPOS (Loss summation) | 196 | 36.9 | 37.2 | 39.2 | 40.9 | 32.0 | 33.0 | 36.0 | 37.0 |
| Morpho + XPOS + GradVacc | 196 | **37.6** | **38.2** | **41.0** | **42.4** | **32.8** | **33.5** | **37.1** | **38.0** |

Table 5: Impact of target side morphological modeling: **English** → **Kinyarwanda**

| **Setup** | **#Params** **x 1M** | FLORES-200 Dev chrF2 | FLORES-200 Test chrF2 | MAFAND-MT Dev chrF2 | MAFAND-MT Test chrF2 | TICO-19 Dev chrF2 | TICO-19 Test chrF2 |
|---|---|---|---|---|---|---|---|
| Morpho + XPOS + BERT + Backtransl. (**Ours**) | 403 | **53.2** | **53.1** | **58.2** | **61.9** | 48.7* | 50.2* |
| Helsinki-opus-mt (Tiedemann and Thottingal, 2020) | 76 | 35.5 | 36.7 | 34.3 | 37.3 | 27.5 | 27.2 |
| NLLB-200 600M (distilled) (Costa-jussà et al., 2022) | 600 | 45.8 | 45.5 | 50.4 | 52.7 | 44.8 | 46.3 |
| mBART (Liu et al., 2020) fine-tuned on our dataset | 610 | 48.7 | 48.5 | 52.4 | 54.1 | 43.7 | 45.2 |
| NLLB-200 3.3B (Costa-jussà et al., 2022) | 3,300 | 50.6 | 50.9 | **57.3** | 58.6 | **50.0** | **52.4** |
| Google Translate | N/A | 59.1 | 60.0 | 76.6 | 87.5 | 46.5 | 49.6 |

Table 6: **English** → **Kinyarwanda**: Comparison of our large model performance after back-translation in relation to open-source models and Google Translate. chrF2 scores are computed using SacreBLEU (Post, 2018) with 10000 bootstraps for significance testing. Highest scores among open source models (p-value < 0.002) are shown in bold. Overall best scores are underlined. *On TICO-19, our model outperforms Google Translate (p-value < 0.002).

| Setup | #Params x 1M | FLORES-200 Dev chrF2 | FLORES-200 Test chrF2 | MAFAND-MT Dev chrF2 | MAFAND-MT Test chrF2 | TICO-19 Dev chrF2 | TICO-19 Test chrF2 |
|---|---|---|---|---|---|---|---|
| Morpho + XPOS + BERT + Backtransl. (**Ours**) | 396 | 54.6 | 54.8 | **54.7** | **59.2** | 49.4 | 50.1 |
| Helsinki-opus-mt (Tiedemann and Thottingal, 2020) | 76 | 35.4 | 35.1 | 33.7 | 35.1 | 29.6 | 29.7 |
| NLLB-200 600M (distilled) (Costa-jussà et al., 2022) | 600 | 53.1 | 52.3 | 51.1 | 54.9 | 47.7 | 48.6 |
| mBART (Liu et al., 2020) fine-tuned on our dataset | 610 | 43.7 | 43.1 | 44.5 | 46.0 | 38.8 | 38.8 |
| NLLB-200 3.3B (Costa-jussà et al., 2022) | 3,300 | **56.8** | **56.0** | **55.4** | **59.6** | <u>**53.4**</u> | <u>**54.1**</u> |
| Google Translate | N/A | <u>60.0</u> | <u>59.1</u> | <u>57.3</u> | <u>64.0</u> | 51.8 | 52.4 |

Table 7: **Kinyarwanda** → **English**: Comparison of our large model performance after back-translation in relation to open-source models and Google Translate. chrF2 scores are computed using SacreBLEU (Post, 2018) with 10000 bootstraps for significance testing. Highest scores among open source models (p-value < 0.002) are shown in bold. Overall best scores are underlined.

in this case use the same Transformer backbone as the morphological models, but without morphological modeling or BERT attention augmentation.

### 3.3 Results

Results in Table 1 through Table 5 show our ablation study results, evaluating the various contributions. In Table 1, we show the improvement across all three benchmarks from adding proper names data to induce token-copying ability. In Table 2, we evaluate the impact of our attention augmentation scheme. The results show substantial improvement by adding BERT and XPOS attention augmentations. Table 3 confirms the effectiveness of adding bilingual dictionary data to the training set. In Table 4 and Table 5, we find a large performance gap between standard transformer with BPE-based tokenization (BPE Seq2Seq) and our morphology-based models (Morpho), which confirms the effectiveness of our morphological modeling. Finally, in Table 6 and Table 7, we use back-translation and train 400M-parameter models that perform better than strong baselines including NLLB-200 (3.3B parameters for English→ Kinyarwanda, 600M parameters for both directions) and fine-tuned mBART (610M parameters). For English→Kinyarwanda, we achieve performance exceeding that of Google Translate on the out-of-domain TICO-19 benchmark.

## 4 Related Work

Morphological modeling in NMT is an actively researched subject often leading to improvements in translation. However most of this research has focused on European languages. Ataman and Federico (2018) shows that an RNN-based word compositional model improves NMT on several languages. Weller-Di Marco and Fraser (2020) evaluates both source-side and target-side morphology modeling between English and German using a lemma+tag representation. Passban et al. (2018) proposes using multi-task learning of target-side morphology with a weighted average loss function. However, Macháček et al. (2018) does not find improvement when using unsupervised morphological analysers. Our studies differs in that it uses a different morphological representation, that is the two-tier architecture, and we also evaluate on a relatively lower resourced language.

The idea of model augmentation with pre-trained language models (PLM) have been previously explored by Sun et al. (2021), and Zhu et al. (2020) who use a drop-net scheme to integrate BERT embeddings. Also, there have been attempts to model word order relationships between source and target languages (Li et al., 2017; Murthy et al., 2019). Our model architecture provides a more generic approach through the attention augmentation.

## 5 Conclusion

This work combines three techniques of morphological modeling, attention augmentation and data augmentation to improve machine translation performance for a low-resource morphologically-rich language. Our ablation results indicate improvement from each individual contribution. Baseline improvements from morphological modeling are more pronounced at the target side than at the source side. This work expands the landscape of modeling complex morphology in NMT and provides a potential framework-solution for machine translation of low-resource morphologically rich languages.

## 6 Limitations

Our morphological modeling proposal requires an effective morphological analyzer and was only evaluated on one morphologically-rich language, that is Kinyarwanda. Morphological analyzers are not available for all languages and this will limit the applicability of our technique. The proposed data augmentation technique for enabling proper name copying ability works in most cases, but we also observed some few cases where inexact copies are produced. Similarly, even with lexical data added to our training, we still observe some cases of hallucinated output words, mostly when the model encounters unseen words. Finally, the proposed morphological decoding algorithm is slower than standard beam search because of the filtering steps and morphological synthesis performed before producing a candidate output token.

Given the above limitations, our model does not grant complete reliability and the produced translations still require post-editing to be used in high-stake applications. However, there are no major risks for using the model in normal use cases as a translation aid tool.

## References

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, et al. 2022. A few thousand translations go a long way! leveraging pre-trained models for african news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070.

Željko Agić and Ivan Vulić. 2019. Jw300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, et al. 2020. Tico-19: the translation initiative for covid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*.

Duygu Ataman and Marcello Federico. 2018. Compositional representation of morphologically-rich input for neural machine translation. *arXiv preprint arXiv:1805.02036*.

Alan Bundy and Lincoln Wallen. 1984. Morphographemics: Alias: spelling rules. *Catalogue of Artificial Intelligence Tools*, pages 76–77.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation.

Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. 2022. Torchmetrics-measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Emmanuel Habumuremyi and Claudine Uwamahoro. 2006. *IRIZA-STARTER 2006: A Bilingual Kinyarwanda-English and English-Kinyarwanda Dictionary*. Rwanda Community Net.

Sharon Inkelas and Cheryl Zoll. 2000. Reduplication as morphological doubling. *Manuscript, University of California, Berkeley and Massachusetts Institute of Technology*.

Mark Kantrowitz and Bill Ross. 2018. Names corpus, version 1.3.

Itamar Kastner, Matthew A Tucker, Artemis Alexiadou, Ruth Kramer, Alec Marantz, and Isabel Oltra Massuet. 2019. Non-concatenative morphology. *Ms., Humboldt-Universität zu Berlin and Oakland University*.

Guolin Ke, Di He, and Tie-Yan Liu. 2020. Rethinking positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*.

Taku Kudo and John Richardson. 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.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers)*, pages 688–697.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Dominik Macháček, Jonáš Vidra, and Ondřej Bojar. 2018. Morphological and language-agnostic word segmentation for nmt. In *International Conference on Text, Speech, and Dialogue*, pages 277–284. Springer.

Rudra Murthy, Anoop Kunchukuttan, and Pushpak Bhattacharyya. 2019. Addressing word-order divergence in multilingual neural machine translation for extremely low resource languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3868–3873.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Toan Q Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*.

Antoine Nzeyimana. 2020. Morphological disambiguation from stemming data. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4649–4660.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. Kinyabert: a morphology-aware kinyarwanda language model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Peyman Passban, Qun Liu, and Andy Way. 2018. Improving character-based decoding using target-side morphological information for neural machine translation. *arXiv preprint arXiv:1804.06506*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183.

Philippe Remy. 2021. Name dataset. https://github.com/philipperemy/name-dataset.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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.

Zewei Sun, Mingxuan Wang, and Lei Li. 2021. Multilingual translation via grafting pre-trained language models. *arXiv preprint arXiv:2109.05256*.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt–building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2020. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*.

Marion Weller-Di Marco and Alexander Fraser. 2020. Modeling word formation in english–german neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4227–4232.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J Martindale, and Marine Carpuat. 2023. Understanding and detecting hallucinations in neural machine translation via model introspection. *Transactions of the Association for Computational Linguistics*, 11:546–564.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*.

# A Model, training and inference hyper-parameters

Our model hyper-parameters along with training and inference hyper-parameters are provided in Table 8.

| **Hyper-parameter** | **Value** |
|---|---|
| **Model** | |
| Transformer hidden dimensions | 768 |
| Transformer feed-forward dimension | 3072 |
| Transformer attention heads | 12 |
| Transformer encoder layers with BERT | 5 |
| Transformer encoder layers without BERT | 8 |
| Transformer decoder layers with GPT | 7 |
| Transformer decoder layers without GPT | 8 |
| *Morpho-Encoder* hidden dimension | 128 |
| *Morpho-Encoder* feed-forward dimension | 512 |
| *Morpho-Encoder* attention heads | 4 |
| *Morpho-Encoder* layers | 3 |
| Dropout | 0.1 |
| Maximum sequence length | 512 |
| **Training** | |
| Batch size | 32K tokens |
| Peak learning rate | 0.0005 |
| Learning rate schedule | inverse sqrt |
| Warm-up steps | 8000 |
| Optimizer | Adam |
| Adam's $\beta_1$, $\beta_2$ | 0.9, 0.98 |
| Maximum training epochs | 40 |
| **Morphological inference** | |
| Beam width | 4 |
| Top scores $M$ | 8 |
| Cut-off gap $\delta$ | 0.3 |
| Minimum affix probability $\gamma$ | 0.3 |
| Global correlation weight $\alpha$ | 0.08 |
| Surface form score stop gap $log(\beta)$ | 2.0 |

Table 8: Hyper-parameter settings

# B Dataset summary

| **Subset** | **Size** |
|---|---|
| **Parallel sentences** | |
| Jw.org website | 562,417 sentences |
| Rwanda's official gazette | 113,127 sentences |
| **Lexical data** | |
| Iriza dictionary 2006 | 108,870 words |
| Kinyarwanda.net | 10,653 words |
| Manually translated | 8,000 words |
| **Augmented data** | |
| Spelled numbers | 200,000 phrases |
| Copy data (e.g. Proper names) | 157,668 words |
| Code-switching foreign terms | 10,276 terms |

Table 9: Summary of our experimental parallel dataset

# C Morphological decoding algorithm

**Algorithm 1:** Inflection generation helper subroutines

**Subroutine:** `inflectionGroupProbabilities`$(T_s, T_p, T_a, P_s, P_p, P_a, M, G)$
- $W \leftarrow M \times G$ `array`
- $W[i, g] \leftarrow -100, \forall i = 1, 2, ..M, \forall g = 1, 2, ..G$
- **for** $i \leftarrow 1$ **to** $M$ **do**
  - $g_s \leftarrow$ `getInflectionGroup`$(T_s[i])$
  - $W[i, g_s] \leftarrow$ `max`$(W[i, g_s],$ `log`$(P_s[T_s[i]]))$
  - $g_p \leftarrow$ `getInflectionGroup`$(T_p[i])$
  - $W[i, g_p] \leftarrow$ `max`$(W[i, g_p],$ `log`$(P_p[T_p[i]]))$
  - $g_a \leftarrow$ `getInflectionGroup`$(T_a[i])$
  - $W[i, g_a] \leftarrow$ `max`$(W[i, g_a],$ `log`$(P_a[T_a[i]]))$
- **end for**
- $V \leftarrow G$ `array`
- $V[g] \leftarrow \sum_{i=1}^{M} W[i, g], \forall g = 1, 2, ..G$
- $P_g \leftarrow$ `softmax`$(V)$
- **return** $P_g$

**Subroutine:** `filterAndCutOff`$(T, P, g, \delta, \gamma)$
- $T' \leftarrow [\,]$
- $pp \leftarrow 0$
- **foreach** $i \in T$ **do**
  - **if** *(*`getInflectionGroup`$(i) = g$*) and (*$P[i] \geq \gamma$*)* **then**
    - $T'.append(i)$
    - **if** $(pp - P[i]) > \delta$ **then**
      - break
    - $pp \leftarrow P[i]$
- **end foreach**
- **return** $T'$

**Subroutine:** `computeScore`$(T_s, T_p, T_a, P_s, P_p, P_a, \rho[.,.], \alpha)$
- $C \leftarrow |T_s| \times |T_p| \times |T_a|$ `array`
- $tot \leftarrow 0$
- **foreach** $s \in Ts$ **do**
  - **foreach** $p \in Tp$ **do**
    - **foreach** $a \in Ta$ **do**
      - $c \leftarrow \exp(\alpha$ `log`$(\rho[s, a]) +$ `log`$P_s[s] +$ `log`$P_p[p] +$ `log`$P_a[a])$
      - $C[s, p, a] \leftarrow c$
      - $tot \leftarrow tot + c$
    - **end foreach**
  - **end foreach**
- **end foreach**
- $C[s, p, a] \leftarrow C[s, p, a]/tot; \forall s \in T_s, \forall p \in T_p, \forall a \in T_a$ `// Normalize`
- **return** $C$

**Algorithm 2:** Inflection generation

**Input** : Probability distributions returned by the neural network (softmax heads) for stems, POS tags, affix sets ($P_s, P_p, P_a$); probability values returned for each affix (multi-label heads) $P_f$; $M$: number of top items (stems, POS tags, affix sets) to consider; $N$: number of top affixes to consider; number of inflection groups $G$; $\rho[.,.]$: corpus-computed correlation of stems and affix sets; $\alpha$: stem-affix set correlation weight; $\beta$: maximum probability gap for inflection groups; $\gamma$: minimum affix probability; $\delta$: maximum probability gap for stems, POS tags and affix sets.

**Output** : Candidate inflected forms and their scores.

**Subroutine** `generateInflections`$(P_s, P_p, P_a, P_f, \rho[.,.], \alpha, \beta, \gamma, \delta)$
- `/* All argSort(.) calls are in decreasing order */`
- $T_s \leftarrow$ `argSort`$(P_s)[:M]$ `// Up to` $M$ `items`
- $T_p \leftarrow$ `argSort`$(P_p)[:M]$
- $T_a \leftarrow$ `argSort`$(P_a)[:M]$
- $T_f \leftarrow$ `argSort`$(P_f)[:N]$
- $P_g \leftarrow$ `inflectionGroupProbabilities`$(T_s, T_p, T_a, P_s, P_p, P_a, M, G)$
- $G_s \leftarrow$ `argSort`$(P_g)$
- $pp \leftarrow 0$
- $R \leftarrow [\,]$ `// List of inflections to return`
- **foreach** $g \in G_s$ **do**
  - $T_s^{'} \leftarrow$ `filterAndCutOff`$(T_s, P_s, g, \delta, 0)$
  - $T_p^{'} \leftarrow$ `filterAndCutOff`$(T_p, P_p, g, \delta, 0)$
  - $T_a^{'} \leftarrow$ `filterAndCutOff`$(T_a, P_a, g, \delta, 0)$
  - $T_f^{'} \leftarrow$ `filterAndCutOff`$(T_f, P_f, g, \delta, \gamma)$
  - $C_g \leftarrow$ `computeScore`$(T_s^{'}, T_p^{'}, T_a^{'}, P_s, P_p, P_a, \rho[.,.], \alpha)$
  - $L_g \leftarrow$ `argSort`$(C_g)$
  - **foreach** $(s, p, a) \in L_s$ **do**
    - `// Formulate affixes by merging affix set's`
    - `// own affixes and extra predicted affixes`
    - $f \leftarrow$ `affixMerge`$(a, T_f^{'})$
    - `// Call morphological synthesizer`
    - $surface \leftarrow$ `morphoSynthesis`$(s, f)$
    - **if** $surface \neq$ `null` **then**
      - $R.append((surface, s, p, a, f, C_g[s, p, a]))$
  - **end foreach**
  - **if** $(pp - P_g[g]) > \beta$ **then**
    - break
  - $pp \leftarrow P_g[g]$
- **end foreach**
- **return** $R$
- `// The returned inflections will be added to the beam`
- `// for beam search-based decoding.`