Group: LionHart Machine Learning Group

Course: Data Science (Cohort 14)

Module: Module 2, Supervised Learning

Technical Mentors: Nikita Njoroge, William Okomba

Project Title: Text Translation using Neural Networks

Group Members:

Tiffany Ndanu

Cynthia Mwadime

Rehema Owino

Dave Njoroge

Mary Mbugua

Kiprop Amos

TEXT TRANSLATION USING NEURAL NETWORKS

BUSINESS UNDERSTANDING

Business Overview

Text translation is converting a written or spoken text from one language to another. There are five ways of translating text:

1. Word-for-word - This method translates according to a word's literal meaning. It doesn't take into account the various cultural and grammatical differences between different languages. It is ideal for documents such as medical research reports.
2. Literal translation - In this method, words are translated without paying attention to their connotations between them. It focuses on the context and strives to find metaphorical equivalents in the target language.
3. Communicative translation - This reflects the exact contextual meaning of the source text in the target language. It takes into account context, culture, grammar, and semantics. It communicates meaning and is frequently used when translating text containing culture-specific idioms, proverbs, or wordplay.
4. Semantic translation - This method emphasizes the aesthetic value of the source text. It strives to convey the syntactic and semantic structures of the source language in the target language. It closely reproduces the original text in a foreign language, while maintaining context and culture It is more flexible and gives the translator more freedom to be creative.
5. Adaptation - this method uses modification or even total rewriting of source text language to find equivalents in the target language while conveying the same message as the original content. It is mostly preferred when presenting messages or ideas in ways different from the source content to the intended public.source

Famous historians and scholars agree that translation date way back before the Bible. Since the earliest days of human interaction, translation has continued to develop, now more than ever, allowing cross-cultural interactions, trade, globalization of the economy, and sharing of knowledge over time. With the help of translation, the world has become more of a melting pot. This also translates into a more necessary service, spanning different theories, mediums, and civilizations. Source

In this study, we will be translating English to various Kenyan local languages using sample texts from English and the target languages.

Problem Statement

There is no doubt that English is a widely spoken language. Most television and radio stations and other communication houses use English to communicate to the people through their various programs or broadcasting news. However, a good number of the target population speak English as a second language. Being able to translate from English to a vernacular language would mean that a wider population can be gotten compared to using English only to communicate important information.

While passing information, such as what is contained in various articles or newspapers, most reporters on television and radio stations that primarily use vernacular languages have to read and then translate the articles or newspapers so that the population understands what they are talking about. This can be tasking especially when an editor or presenter has a lot of material to pass to the population, specifically articles.

To solve this problem, we will be building a model that can translate English to local languages(Luo). The model shall be integrated with an interactive web application user interface that can take in text input via streamlit. The model can be expanded to translate other local languages.

Justification

Several translation websites mostly translate between international languages such as English to Swahili. In Kenya, there a professional bodies that offer translation and interpretation services(source). Hiring these services can be quite expensive, especially when trying to communicate a piece of important information such as constitution interpretation to a predominantly native-speaking community. Having a web application can greatly reduce the burden of outsourcing translation services every time they are needed.

Main Objective

To provide a platform where Kenyans can get a translation of languages from other communities.

Specific Objectives

- Create a TensorFlow model that can translate English to a Kenyan native language.
- To determine the most accurately translated language given the model's accuracy.
- Create an interactive web application user interface that can take user input in English and provide a translation in the selected vernacular language.

Research Questions

- Which language is the most accurately translated?
- Can a TensorFlow be used to translate local languages with a high level of accuracy?
- Can an interactive interface perform as expected in taking in input and providing appropriate language translation?

Project Plan

We shall use the CRISP-DM Methodology for project planning.

We shall use the JIRA Kanban board for this project management.

We shall use google collaboratory notebook as our coding environment, Python programming language, and the TensorFlow library for training and making inferences on deep neural networks.

We shall use the pyplot module from the matplotlib library to create visualizations if any.

We shall use streamlit for deployment.

We shall prepare a PowerPoint presentation for the presentation of our findings.

Assessing the Situation

Resource Inventory

The data we will be using is a collection of random words, phrases, and sentences in English and their respective translations in Luo. Can be found here.

Assumptions

The data is correct, that is, the translations from English to Luo are correct.

Constraints

There were no constraints.

Data Mining Goals

- Create an interactive user interface via a web application where a user can type an English sentence and get a translation in the selected native language.
- Build a model to translate an English sentence into a selected Kenyan native language.

DATA UNDERSTANDING

Data Understanding Overview

We had one dataset with one table.

Data Collection

The data was obtained by randomly translating various English words, phrases, and sentences into their respective Kalenjin translations and stored in an Excel spreadsheet. Some more data was collected from biblical texts in both English and Kalenjin and stored in text files.

Describing the Data

The table in our dataset has 2 columns and 300 rows. The first column contains the English words, phrases, and translations while the second column contains the respective Kalenjin translations.
The data type for both columns is a string.

Data Exploration

At this point conducting an initial exploration of the data using visualizations such as tables, charts, and other visualization tools is quite difficult on the raw translations. The data has to be pre-processed first in the data preparation phase to produce data that can be meaningfully explored.

Data Quality Validation

- Missing data - the table has no blank spaces or data coded as non-response(null, ? or 0).
- Data errors - no typographical errors were made during the data entry exercise.

- Measurement errors - the data is not using any data that requires any measurement schemes.
- Bad metadata - in both columns, the apparent meaning of the fields matched the meaning of the field name. In this case, the English column has words and phrases in the English language and the Kalenjin column has words and phrases in the Kalenjin language

DATA PREPARATION

The following steps were carried out during the data preparation phase:

Installations

We first started by installing the following necessary libraries and modules:

1. JPype1 - this module allows us to access Java within Python. It allows Python to make use of Java only libraries, exploring and visualization of Java structures, development, and testing of Java libraries, scientific computing, and much more.
2. Aspose Cells for .NET - this class library allows us to manipulate and process spreadsheet files within our applications. Combined with APIs and GUIs, it speeds up MS Excel programming and conversion.
3. TensorFlow text - this provides a collection of text related classes and ops. The library can perform the pre-processing regularly required by text-based models and includes other features useful for sequence modeling not provided by the core TensorFlow.

Importing the necessary libraries

Then we imported the following libraries:

- TensorFlow for working with input in text forms such as raw text strings or documents.
- Pandas library for data analysis and manipulation
- Numpy library for numeric computations
- Matplotlib and seaborn for visualization
- Regular expressions library(re) for data preprocessing.

Loading the dataset

The dataset was loaded from the excel file into our working environment using the Pandas library.

Pre-processing the Data

The following steps were taken to prepare the data the for text translation:

1. Creating a tf.data dataset

This is done to ensure that the array of stings loaded is shuffled and batched efficiently for the model.

2. Standardization

Using the tensorflow_text package, we performed Unicode normalization to split accented characters and replace compatibility characters with their ASCII equivalents. All the punctuation marks were removed in this part and the text is converted to lower case.

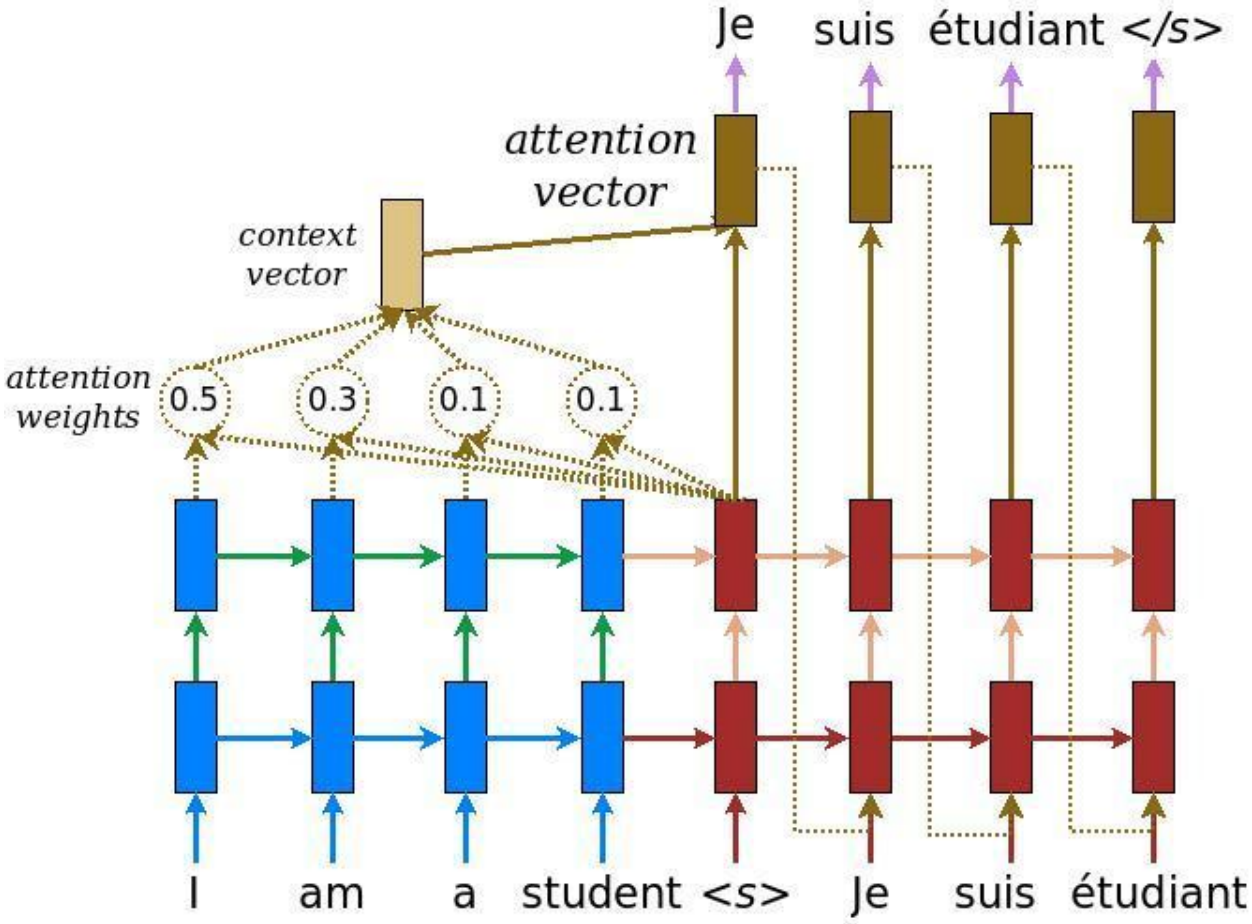
3. Text Vectorization

In this step, we get the vocabulary of the data. The vocabulary determines the complexity of the problem i.e the more complex the vocabulary, the more complex our problem. This was accomplished by wrapping the standardization function in a Keras text vectorization layer which handles the vocabulary extraction and conversion of the input text to a sequence of tokens.

MODELING

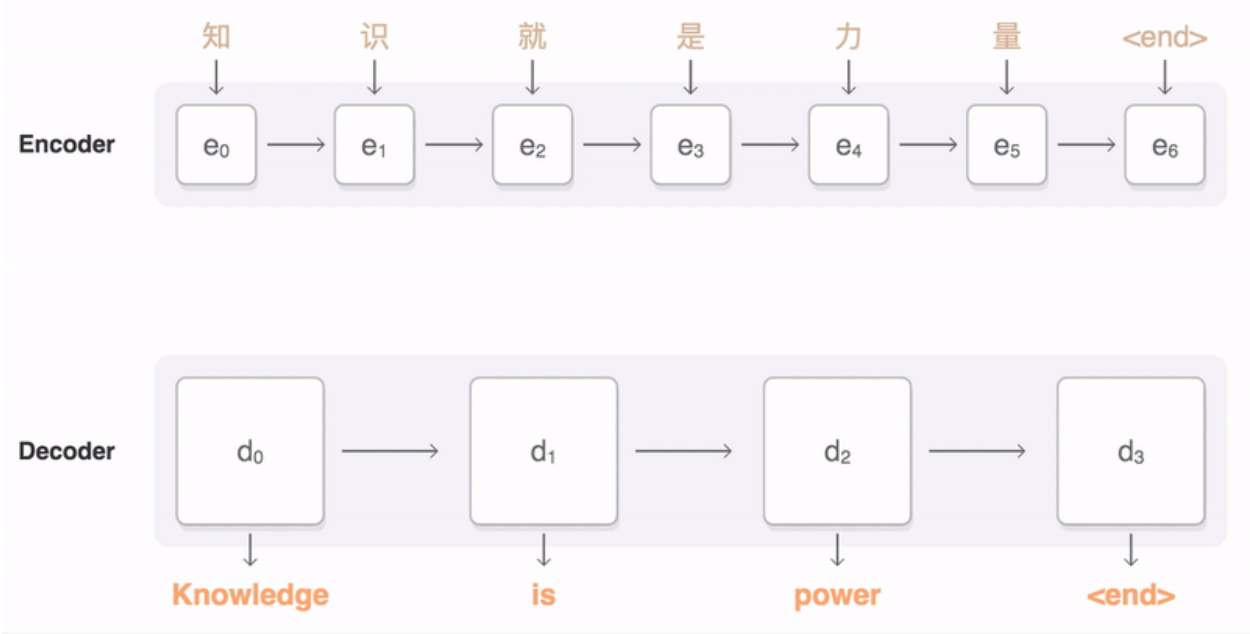
Selection of Modeling Technique

We decided to use the sequence-to-sequence (seq2seq) modeling technique to translate English to Luo. This technique is a little bit outdated but it is still very useful to get a deeper understanding of attention mechanisms. In this technique, both the input and output are sentences, that is, the sentences are the sequences of words going in and out of the model. The following diagram shows an overview of the seq2seq technique [source]:



This technique has 2 major components: the blue part represents the decoder and the red part represents the decoder.

The seq2seq was introduced together with neural attention models that enable neural networks to become more selective about the data they are working with at any given time. The core focus of the neural attention mechanism is to learn to recognize where to find the important information. Here's is an example of a neural machine translation: source



The cycle works as follows:

1. The words from the input sentence are fed into the encoder to deliver the sentence meaning (the thought vector)
2. Based on the thought vector, the decoder produces words one by one to create the output sentence
3. Throughout the process, the attention mechanism helps the decoder to focus on different fragments of the input sentence

Building the Model

The encoder/decoder model:

The encoder takes in sentences that were preprocessed in form of token IDs from the preprocessing step. These tokens are embedded to produce a new sequence that is passed

to the attention head. Together with the new sequence of sentences is an initial state for each of the sentences that are used to initialize the decoder.
The sequences generated from the encoder pass through the attention head to the decoder. This is because the decoder uses attention to selectively focus on parts of the input sequence. The attention takes a sequence of vectors as input for each example and returns an "attention" vector for each example.
At the decoder, we get to see the generated predictions for the next output token. This is achieved in the following steps:

1. It receives the complete encoder output.
2. It uses an RNN to keep track of what it has generated so far.
3. It uses its RNN output as the query to the attention over the encoder's output, producing the context vector.
4. It combines the RNN output and the context vector using this equation to generate the "attention vector".
5. It generates logit predictions for the next token based on the "attention vector".

Parameter Settings

These are the parameters used in building the model:
Batch size = 16
Epochs = 37
Embedding size/dim = 512
Units = 1024

Running the Model

This was a pretty straightforward task. After plugging in the parameters, the model is executed and observed for any viewable results. The parameters were tweaked several times to see different outputs.

Model Assessment

The neural network using the translation with attention(seq2seq) had a difficult time learning the new language, Kalenjin, and it found it difficult to translate the English sentences to meaningful Kalenjin sentences.
With time and a lot more data and training time, we were able to achieve greater accuracy from the model and more precise predictions. The following are some of the snippets from the model's output:

```
'kiamwaite eng kutinnyu kiruogutik tugul che bunu kutingung', # "with my lips have i declared all the judgments of thy mouth
'a kiprutoiyo eng ngony ameungena ngatutiguk', # "i am a stranger in the earth hide not thy commandments from me"
```

Sample input text and expected output

```
i have recited aloud all the regulations you have given us .
i am only a foreigner in the land . dont hide your commands from me !
```

Sample output - the expected output has not been achieved, but it is close tp what we expected.
The results of the translation attempt are as follows:

1. Sentence BLEU Score - 61.47%. This implies the sentences are relatively comparable.
2. Corpus BLEU Score - 61.47%. This implies that the translation maintaned its context.A score of 0.6145 indicates that the Quality of the translation is often better than human.
3. Individual N-Gram Scores - 14.28%. A very low n-gram score was achieved. The original text had six words but the the translation had fourteen letters. This shows the model isn't fully optimized and is still bunching together alot of similar words and needs to be optimized further.

EVALUATION

Evaluating the Results

The overall results of the first attempt at translating English to a Kenyan local language are quite easy to communicate from a business perspective. The study did produce what we hoped for: an algorithm that can take in English words and phrases and output the respective Kalenjin translations. Using this algorithm, we can continue training more data to improve the accuracy of the model. We can also train other local languages and get their translations using the algorithm.

New Questions: From the study, the most important question to come out from it is, how far is too far? How much more data can the algorithm take to keep learning while preventing overfitting?

Review Process

We had no initial algorithm to compare our algorithm with except the ones that translate international languages like the Google English - Swahili translator. But applying the CRISP-DM process in our study helped to appreciate the cyclic nature of the process and to understand that it increases power. The CRISP-DM process helped us understand that:

- A return to the exploration process is always warranted when something unusual appears in another phase of the CRISP-DM process.
- Data preparation requires patience, since it can take a very long time.
- It is vital to stay focused on the business problem at hand because once the data are ready for analysis, it's all too easy to start constructing models without regard to the bigger picture.

Conclusion

There are already websites and applications that attempt to translate English to Kenyan vernacular languages. Some of these websites are not as efficient as they supposedly advertise. Talk about wrong translations, misuse of words while forming or giving sentence examples, the list can go on and on.

Having a website that does this job perfectly can be a dream come true for a lot of people looking to translate English to a local language for whatever reason. Our model has proved to be a more effective way of proving translations on a web application. It is, therefore, more time effective and cost-efficient compared to hiring translators.