Machine Learning Engineer Nanodegree

Capstone Proposal

Daniel Lameyer
April 27, 2019

Proposal

Domain Background

Deep Learning technology has exponentially expanded the possibilities of machine learning capabilities in recent years. One of the most popular applications of which is in natural language processing and language translation due to the technology's ability to process the more complicated structure that is human language. Thanks to advancements in this field, in 2016, Google announced that its popular Google Translate services will transition to artificial neural network based algorithms as the foundation of its translation software. Among the various deep neural network architectures, the Recurrent Neural Network is structure is commonly used for NLP tasks due to the temporal & sequential structure of languages. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

In this project, I propose to create a machine learning model that can translate a Japanese sentence to English. Translating between Japanese and English is notoriously difficult due to the vast linguistic differences in the structure, grammar, and vocabulary of the languages. Languages with similar roots or linguistic history are often easier to translate to each other both by humans and machines. However, dissimilar languages often have an added challenge. I am bilingual in these two languages, and I hope to enter a career in NLP engineer working with Japanese and English. I hope to use this project as a foundation in developing my multilingual NLP skills.

Problem Statement

The objective of this project is to build a model, learned using data samples of Japanese sentences with their English translations, that will automatically return the translated English sentence when a new Japanese sentence has been passed to it. The model will take in a Japanese sentence (with spaces), tokenize it, and return an English sentence that reflects or maintains the general meaning of the original sentence.

Datasets and Inputs

The dataset for this project will utilize the corpus provided by Yusuke Oda which is a sample of the Tanaka Corpus, filtered to sentences with 4~16 words and pre-tokenized. The corpus contains a training dataset of 50,000 Japanese/English sentence pairs, each English line has been human translated from the original source. This corpus is perfect for the proposed project, for one the human translated sentence provides the most natural labels the model can compare for training and testing purposes, and the dataset already contains pre-tokenized data which is very helpful for the Japanese corpus since the language does not contain space delimiters in sentences between words much like the English language does.

Solution Statement

At its core, this model will be a supervised machine learning algorithm. Therefore, the output of the model can be compared against a label value which the model should have produced. However, since a language translation is not a simple binary or multi-label classification but a collection of words that produce meaning, using standard metrics such as accuracy, precision or recall may not suffice. Instead in order to provide a metric for the performance of the translation, I propose utilizing the Levenshtein distance to calculate a Word Error Rate to compare how close the output matches the label sentence. The Levenshtein distance between two strings is measured as the number of character changes needed to convert to the other string. The validation dataset from corpus provided by Yusuke Oda will be used to measure the accuracy of each translated output to the validated translation.

Benchmark Model

Much like Google's Neural Machine Translation System the basis of this model will be a deep neural network with an RNN architecture. Recurrent Neural Networks are a very common architecture for NLP machine learning tasks due to the sequential nature of human languages. The RNN may contain additional hidden layers and bidirectional architectures due to the complexity of the linguistic difference between Japanese and English. For a bench mark RNN model, the author will aim for a validation score of 97.5% achieved by Thomas Tracey in his English to French translation model.

Evaluation Metrics

The basis of the evaluation will be to compare the English sentence the model produces, against the appropriate label sentence which the model was intended to produce. Since translation can be a very difficult concept to score with multiple methods to evaluate, there are numerous options to be selected here. The proposed evaluation metric is to utilize the Word Error Rate to evaluate a score between the model output and target sentence.

The Word Error rate calculation is as follows: WER = (S+D+I) / N = (S+D+I) / (S+D+C)

Where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, and N is the number of words in the reference (N=S+D+C). This methodology provides a simple yet effective method to determine how closely a sentence resembles another.

It is important to take note that one shortfall of this method is that the semantics of the original sentence is not taken into considerations. Although word selection may closely resemble the target sentence, words out of order or small differences in diction can vastly alter the intended meaning of the original sentence.

Project Design

At a high level, the project will proceed with the following steps to accomplish creating the machine translation model.

1. Importing and Loading training data sets
2. Build the Preprocessing Pipeline for the Japanese and English Sentences
3. Architect the Deep Learning Model & Train on the Data

4. Create the Evaluation pipelines
5. Evaluate on a Validation Dataset

First, the training data for the model will be imported from Yusuke Oda's corpus. The document of 50,000 Japanese sentences and their English translations will each be imported into a list variable. Then there will be some preliminary exploration of the data to view sample sentences and vocabulary to determine whether the dataset contains adequate translations and samples. Note, some readers of this project may not be able to judge whether the translations are adequate or not if they do not understand Japanese. Here is a sample of the translated sentences provided in the corpus:

Line 1:

JP: 誰 が 一番 に 着 く か 私 に は 分か り ま せ ん 。

EN: i can 't tell who will arrive first .

Line 2:

JP: 多く の 動物 が 人間 に よ っ て 滅ぼ さ れ た 。

EN: many animals have been destroyed by men .

Line 3:

JP: 私 は テニス 部員 です 。

EN: i 'm in the tennis club .

After the data has been imported, the corpus sentences will need to be preprocessed in preparation for the deep learning algorithm. Preprocessing steps include tokenizing the dataset to breakdown each sentence to its tokens(words) and padding the data which will fill in space for shorter sentences with empty values so that the input data can be passed to the model expecting a certain size input. These preprocessing steps will prepare the Japanese and English datasets to be passed to the neural network.

The deep learning model will at its core be a recurrent neural network model. To accomplish this, the Keras package will be utilize to quick develop a neural network architecture with the appropriate parameters and layers. Once an architecture is made, the model will begin to train off the dataset, learning the English translations of the Japanese input dataset. As it trains, the progress of the model can be observed as the model will print out validation loss and accuracy scores as it trains. Several attempts will most likely be required during this step to fine tune the model to produce better performance from the training step.

As outlined earlier, the Word Error Rate method will be used to evaluate the output of the model. To do that, a pipeline that will take the English sentence produced by the model will need to be compared against the target translation. In this step a series of functions will be created to easily score the output sentences with the label value using the WER formula.

Once the evaluation pipeline has been created, the validation dataset will be used to pass on to the model to evaluate its performance. This validation dataset is provided by Yusuke Oda's corpus and does not overlap with any sentences in the training dataset. The Japanese sentences will be passed to the model, yielding an English sentence. Then the English sentence will be passed through the evaluation pipeline to compare against the label sentence. The pipeline will produce a score as to how close the output sentence resembles

the target translation. This process will be repeated across the remaining validation dataset, and an average score will be calculated to determine the overall performance of the model.

References

1) Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

2) Word Error Rate

3) Language Translation with RNNs: Build a recurrent neural network that translates English to French