University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

# Neuro MT: A Learning Machine Translation Web Service

A special problem in partial fulfillment
of the requirements for the degree of
**Bachelor of Science in Computer Science**

Submitted by:

Alfred John G. Tacorda

June 2017

Permission is given for the following people to have access to this SP:

| | |
|---|---|
| Available to the general public | Yes |
| Available only after consultation with author/SP adviser | No |
| Available only to those bound by confidentiality agreement | No |

## ACCEPTANCE SHEET

The Special Problem entitled "Neuro MT: A Learning Machine Translation Web Service" prepared and submitted by Alfred John G. Tacorda in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

**Marvin John C. Ignacio, M.Sc. (*cand.*)**
Adviser

**EXAMINERS:**

| | **Approved** | **Disapproved** |
|---|---|---|
| 1. Gregorio B. Baes, Ph.D. (*cand.*) | _____ | _____ |
| 2. Avegail D. Carpio, M.Sc. | _____ | _____ |
| 3. Richard Bryann L. Chua, Ph.D. (*cand.*) | _____ | _____ |
| 4. Perlita E. Gasmen, M.Sc. (*cand.*) | _____ | _____ |
| 5. Ma. Sheila A. Magboo, M.Sc. | _____ | _____ |
| 6. Vincent Peter C. Magboo, M.D., M.Sc. | _____ | _____ |

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

**Ma. Sheila A. Magboo, M.Sc.**
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences
and Mathematics

**Marcelina B. Lirazan, Ph.D.**
Chair
Department of Physical Sciences
and Mathematics

**Leonardo R. Estacio Jr., Ph.D.**
Dean
College of Arts and Sciences

**Abstract**

Communication through words is the most effective way of exchanging information between people of the same language. But it may prove difficult if there's a language barrier. The advent of computers provided a solution that is an immediate and reliable substitute to a dictionary or a translator, the solution is called Machine Translation(MT). The problem with the common implementation of MT, Statistical Machine Translation(SMT), is that it requires discrete models that derives a part of the translation before learning a translation. The usage of Neural Machine Translation (NMT), removes the need to rely on discrete models to learn a translation. This special problem uses NMT for Filipino to English bidirectional translations. It also provides a web service that is easily integrable to mobile and web applications. The system also allows users to submit translation suggestions. This allows the NMT system to be improved with new data. The trained NMT models included in this special problem produces translations that are comparable against the current SMT used for Filipino to English bidirectional translations.

*Keywords*: Neural Machine Translation, Web Service, Neural Networks, Artificial Intelligence, Machine Translation

# Contents

# List of Figures

# List of Tables

# I. Introduction

## A. Background of the Study

Communication through words is the most effective way of exchanging information between people of the same language. A figurative phrase that describes the difficulty of communicating with no language in common can be referred to as a language barrier. For years, we have crossed this barrier by using a dictionary, or having a bilingual, a person who is fluent in two languages, translate for us[1]. The advent of computers provided a solution that is immediate and reliable substitute to a dictionary or a bilingual. This solution of translating an input sentence from one language to another is called Machine Translation(MT).

MT started as an automated word-per-word conversion through the use of a bilingual dictionary. With a set of defined rules, it can also form proper sentences[2]. This method can be likened to Artificial Intelligence(AI) which is programmed with a set of rules that specifically handles each task it needs to perform[3]. However, this method is restricting because you need to define all the word translations and rules to have an adequate translation from one language to another. We need a method that automatically learns these conversions without explicitly defining all the rules. Machine learning(ML) is a type of Artificial Intelligence(AI) that provides computers the ability to learn without being explicitly programmed[4]. By applying some concepts of ML, a new method called the Statistical Machine Translation(SMT)[5] emerged.

SMT is an approach which finds the most suitable translation of a sentence by searching for its most frequent translation[5]. Additionally, it learns patterns of a certain translatio, by simply defining a set of parameters that controls the translation rules and a large set of sentence pairs of two different languages[5]. Today, SMT is the most used MT solution because it is easily accessible through the application

called Google Translate[6]. Another application that is prominent among linguists and researchers is called Moses. It allows them to easily redefine translation rules and update the dataset of sentence pairs which allows more room for improvement in SMTs[5]. Research institutes such as the Philippine-California Advanced Research Institutes(PCARI) not only researches on improving a MT approach, but also allows sponsored Filipino students to learn from their research by providing them the tools and data.

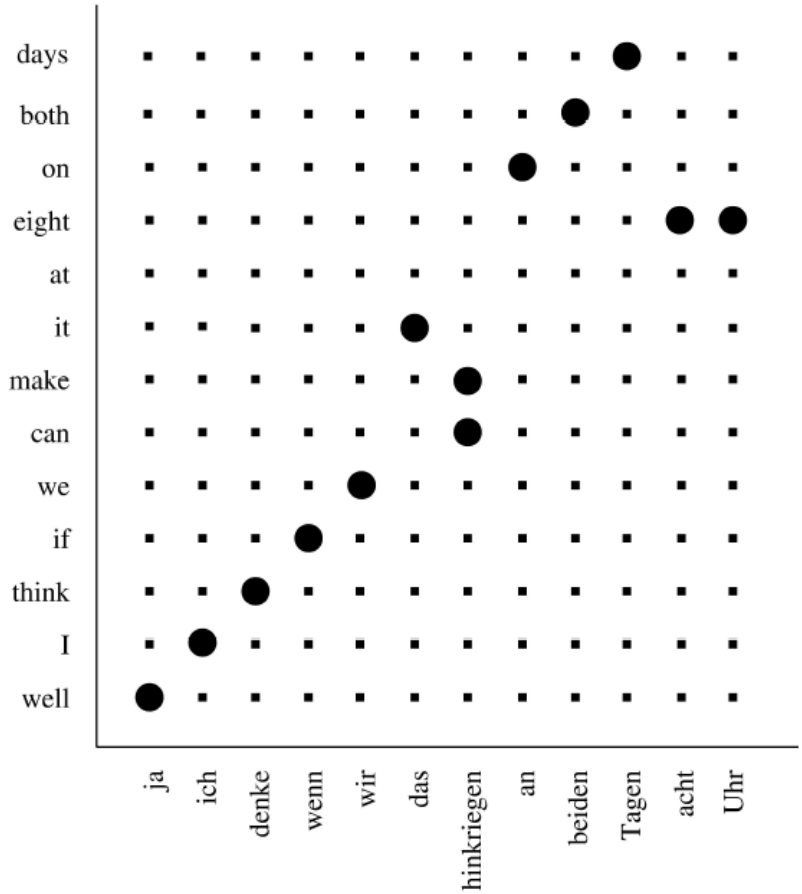


Figure 1: An example of the alignments learned by an IBM Model

The process of training a Baseline SMT model using Moses is simple. Given a set of sentence pairs from two languages, an International Business Machines(IBM) Model finds the word alignments per sentence pair. Alignments point the expected position of the source word in the target sentence. From those alignments Moses forms word

pairs. The next step would be getting a lexical translation table. Given the word pairs, it is possible to estimate a maximum likelihood from the lexical translation table. After that, Moses extracts phrases from the previous data. The extracted phrases are then scored. The phrase scores will be the basis for choosing the most probable translation. The second to the last step is building a reordering model, it determines how the groups of translated phrases or words will be reordered. Finally, the generation model computes the probabilities for the given translation direction. To train a SMT Model that considers part of speech, lemmas, surface form, etc. in the training data, a type of training called Factored Traning is needed to be performed. This training also requires the previous steps mentioned.

SMT is still improving until today, but it derives its translations from discrete word pairs in the source and target languages[7]. That discrete representation of word pairs will be its limit. It can use more concepts from ML like Neural Networks(NN), which represents its learning of a task continuously to further extend its limit. NN is a system which is modeled on the human brain. The neurons of the human brain allow it to process complex data and generalize patterns from it[8]. Like the human brain, NN is composed of neurons[8]. However, we need to design the structure of neurons for it to properly generalize patterns from the huge data provided to it[8]. The approach of Cho et al.[9] used Recurrent Neural Networks(RNN) to get the phrase scores out of the set of phrase pairs extracted by Moses. A RNN is a neural network designed to handle sequences of data such as sentences. The experiment performed by Cho et al. resulted to an improvement of the translation score of the SMT. It is because the phrase pairs were evaluated in a continuous model, an RNN. With this in mind we can replace more models of SMT into a continuous model for it to further improve. But a problem will arise from that. We still need to improve each of the continuous models for the whole translation to further improve. Therefore, we have to find a solution where we only need to provide the data and design how the data

will be evaluated by the system, as a whole. An approach like that exists, and it is known as Deep Learning(DL) [10]. Through the technique of DL, another study of MT emerged, it was termed as Neural Machine Translation(NMT)[11].

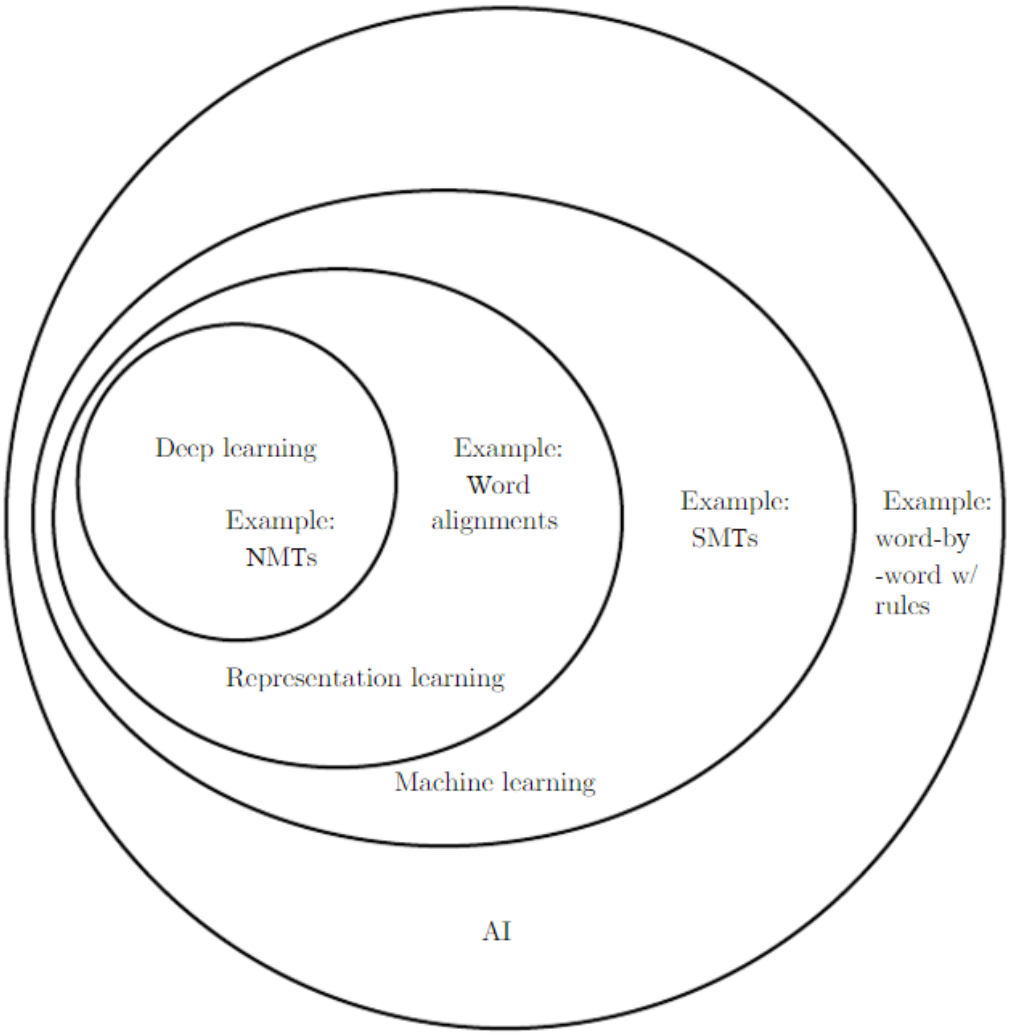


Figure 2: A Venn diagram showing the hierarchy of each method and an example for Natural Language Processing

NMT only requires large set of sentence pairs to learn the translation from one language to another. It works that way because DL allowed researchers to build deeper neural networks[10, 11]. This resulted to a neural net that can be designed in such a way that can handle more than 1 task. Features are parts of an input where an NN looks into. It gives clues on how the input will be evaluated. An example of a

this is sentence alignments. An AI expert does not need to define the parameters of a translation task, but it needs to declare hyperparameters to define how the NMT model learns the task. The goal of this learning task is to reach the optimal weights of each neuron for a certain translation task. The hyperparameters are epochs, learning rate, batch size, and solver type. Epochs define how many times the whole training data will be processed by the model. Learning rate defines the speed on how fast the initial random weights reach the optimal weights. A high learning rate can cause the current weights to skip the optimal weight, while a low one may take too long to reach the it. Batch size defines how many sentences can be processed by the model before it adjusts the weights. Solver type is an optimization algorithm that adjusts the weight based on the errors accumulated in a given batch.

With DL, NN's learned the translation and sentence alignments using only a single architecture[11]. The proposed model by Bahdanau et al.[11] added a mechanism inside the neural network to learn these alignments. This eliminates the IBM Models from being a part of deriving translations and allows the whole architecture to be jointly improved. Additionally, it was proven that in nearly all 30 translations performed in a recent case study by Dowmunt et al.[12], that NMTs are just as good or better than Phrase-Based SMT in terms of quality. Bilingual Evaluation Understudy(BLEU) score is the metric used to measure quality. It measures how close a MT is to a human translation[13]. A score of 100 is a perfect score while 0 is the worst result. ASEAN MT[14] achieved a score of 38.12 BLEU for English-Filipino and 35.79 BLEU for Filipino-English. Nocon et al.[1] achieved a score of 31.15 BLEU for English-Filipino and 32.71 BLEU for Filipino-English. Languages covered in the case study of Dowmunt are Arabic, Chinese, English, French, Russian, and Spanish. These languages have varying complexities and are unique in their own way. With that in mind, English to Filipino bidirectional translations using NMT is a possible study.

"Communication between different nations is essential. Languages which are foreign to another impose difficulty in understanding" said by Nocon et al.[1]. When trying to engage in conversations to an unfamiliar language, people cannot always rely on human translators because of it being too costly. As an alternative, they tend to use online sites such as the MT service created by the ASEAN MT[14], Google Translate, IBM Watson[15], or SYSTRAN[16] for translations. ASEAN MT and Google Translate are both Phrase-Based SMT's that supports English to Filipino bidirectional translations. IBM Watson is also an SMT but it does not support English to Filipino bidirectional translations. SYSTRAN is an NMT that supports more than 30 languages worldwide[16]. IBM Watson and Google Translate has a paid web service which can be integrated to any web and mobile application[17, 15]. SYSTRAN has a paid application that provides desktop and office plugins[16]. Both Google Translate and SYSTRAN has a web browser plugin that is used to translate web pages.

## B. Statement of the Problem

The problem with Phrase-Based SMT systems is that in all languages it uses discrete models to learn a translation. This can affect the semantical and syntactical representation of the translation of SMT. Furthermore, the problem with an SMT that replaced each of its parts with a Neural Network to become continuous, is that you have to improve each of the models separately to improve the translation. The implementation of ASEAN MT cannot be integrated like a plugin to any portable application which is frequently used by a Filipino or a foreigner. While SYSTRAN, an NMT, does not support Filipino to English bidirectional translations.

## C. Objectives of the Study

### C..1 General Objective

To use NMT to provide semantically and syntactically meaningful Filipino to English bidirectional translations, to provide a web service that is easily integrable to mobile and web applications, and to allow the NMT system to be improved with a new architecture or data.

### C..2 Specific Objectives

1. Create a mobile and web application that allows users to
    (a) Translate English sentences to Filipino sentences.
    (b) Translate Filipino sentences to English sentences.
    (c) Suggest translation correction.
2. Create a web service that allows developers of a web or mobile application to
    (a) Use the Web Service Application Program Interface(API) to
        i. Request translations.
        ii. Get translations.
        iii. Send translation correction.
3. Allows an AI Expert to
    (a) Prepare the dataset
        i. Enter Training Name
        ii. Select the training dataset of English to Filipino sentence pairs
        iii. Select the validation dataset of English to Filipino sentence pairs
        iv. Display the number of training and validation data

v. Tokenize the dataset

vi. Learn the truecasing of the dataset

vii. Learn and apply Byte Pair Encoding

viii. Build the dictionary

(b) Create a NMT model

i. Use a prepared dataset

ii. Select a NMT architecture

iii. Select hyperparameters

A. Optimizer or Solver type (adadelta, stochastic gradient descent, adam, rmsprop)

B. Embedding Layer Size

C. Hidden Layer Size

D. Learning rate

E. Source Vocabulary Size

F. Target Vocabulary Size

G. Minibatch Size

H. Validation Minibatch Size

I. Display Frequency of Loss

J. Validation Frequency using BLEU Score

K. Save Frequency

L. Sample Frequency

M. Max Minibatches

N. Max Epochs

iv. Select additional mechanisms

A. L2 Regularization Penalty

B. L2 Regularization Penalty(Original Weights)

C. Alignment Regularization

D. Gradient Clipping Threshold

E. Dropout for Input Embeddings

F. Dropout for Hidden Layer

G. Dropout for Encoder

H. Dropout for Decoder

v. Train the architecture to create a NMT model

vi. Display training and validation error for each epoch

(c) Evaluate the NMT model

i. Select the model to evaluate

ii. Perform multiple predictions using a test dataset

A. Select the test dataset

B. Preprocess the test dataset

C. Translate all the test data using the model and output to a log file

D. Display the BLEU score for the whole data

iii. Perform a single prediction using a single sentence

A. Input a single sentence

B. Translate the sentence

iv. Deploy the evaluated model

4. Allows a Language Expert to

(a) Save new translation suggestions.

(b) Update dataset.

(c) Remove dataset.

## D. Significance of the Project

The project provides a bidirectional English to Filipino NMT system which is easily integrable. Any application that integrated this system can provide immediate translations anytime and anywhere to a local, foreigner, researcher, or student that frequently uses the it. The system can train new models which may be better or worse in translating than the current trained model. Moreover, the system accepts and stores translation suggestions from its users, which in turn can help the system improve after retraining. It can also help social researchers mine location-specific translation data. The project could also be used as reference for Neural Machine Translation development in the Philippines.

## E. Scope and Limitations

1. The trained model will perform best in formal English to Filipino bidirectional translations.

2. This project uses data of around 120,000 English and Filipino sentence pairs which was provided through the PCARI project entitled E-Participation 2.0: Connecting Diverse Philippine Populations for Disaster Risk Management with a Toolkit Integrating Text and Speech Analytics (project no. IIID 2015-07).

3. The get translation request is a JSON request, requested in the location provided by the web service.

4. The request translation request is a JSON request with source language, target language, and input sentence as its parameters.

5. The suggest translation request is a JSON request with the input sentence and target sentence. Each sentence is contained in the variable of the specific language.

## F. Assumptions

1. The input is assumed to be US English with proper grammar and sentence structure for English to Filipino translations.

2. The input is assumed to be Filipino with proper grammar and sentence structure for Filipino to English translations.

3. The sentence input will not exceed 80 words.

4. There would only be one language expert.

# II. Review of Related Literature

There are SMT's that support Filipino to English bidirectional translations. One work by Nocon et al.[1] which used Moses achieved a BLEU score of 32.71 for English-Filipino translations, while in Filipino-English it got 31.15. Accepted systems of NMT and SMT achieved scores ranging from 20 to 40 in most languages. One cause for not reaching a score near 40 is because of out-of-vocabulary words or OOVs. These are input words not included in the list of phrase pairs. Another system by Ang et al.[18] also supported Filipino to English bidirectional translations. The system added a function where a feedback from a user can be readily used by Moses. The idea is that the feedback will serve as a correction to the translation and the system is expected to learn from it. However, incorrect feedback may ruin it. The system with human feedback achieved a better BLEU score compared to without human feedback only on the first few iterations of the test. The BLEU score gradually decreased as the iteration increased.

SMT is still improving until today. It is easy to design and implement but hard to improve because you need to engineer the features unlike NMT. One way to harness the specialties of both SMT and NMT is by combining it. A work by Devlin et al.[19] combined Neural Networks and SMT, it is called Neural Network Joint Models (NNJM). This implementation improved most SMT systems' BLEU score to a range of around 3.0 to 6.0. Another method which combined SMT's with NN is by Cho et al.[9]. It used Recurrent Neural Networks(RNN) for scoring the phrase pairs which are used by the SMT to translate. RNN is a deep neural network that is designed to generalize from a sequence of data like sentences. By adding RNN the system improved by 0.57 BLEU on their tests.

NMT systems can learn the features of a sentence. The proposed approach of Cho[9] learned the phrase representations of translations. Another approach is by Bahdanau et al.[11], it learned to align the words of a sentence from one language to

another. These alignments learned are also the alignments required by an SMT for it to learn a translation. This approach uses the proposed model of Cho[9] but adds an attention mechanism which learns the alignments of the words of translations from one language to another. Additionally, this approach solved the problem in Cho's model of having difficulty in translating long sentences with length of 20 or more words[11]. Another approach is by Sutskever et al.[20], it used Long Short-Term Memory(LSTM) an RNN with a memory which is better at memorizing relations of inputs far away from each other in a sequence. This means that given a sentence, LSTMs can see the relation of two words that are far apart. This resulted to a score comparable to the best SMT that time.

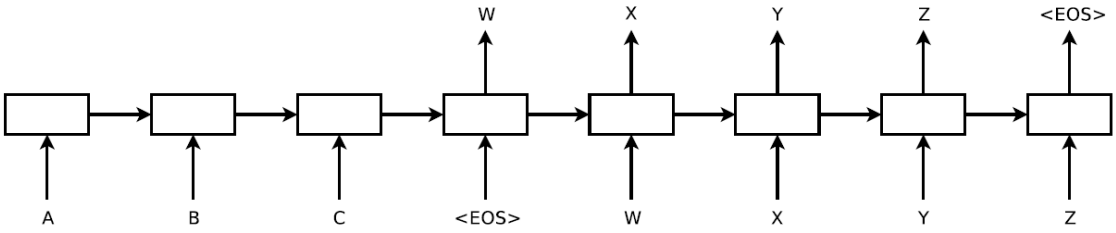


Figure 3: Sequence to Sequence learning with LSTMs

NMT can compete against SMT but it is has a problem with unknown words or OOVs. Unknown words are words in a form like $< unk >$ which indicates that the word translation is not known by the system. An NMT system is usually limited to a number of words known called vocabulary since using the whole vocabulary of a certain language will make it too long, this adds complexity to the operations that NMT will perform. In the case of SMT, OOVs occur because it cannot find the target word, it doesn't exist on the phrase pairs[1]. One approach that solves this problem is by Luong et al.[21]. This approach used the same model as Sutskever's[20] but it added an alignment mechanism which points the produced unknown words of the translated sentence to the each corresponding word from the input sentence. Each pointed word was then translated using a dictionary and it is used to replace

the unknown word in the translated sentence. Sennrich et al.[22] achieved an open-vocabulary translation by encoding rare and unknown words as sequences of subword units. They adapted Byte Pair Encoding(BPE) to produce a fixed-size vocabulary of variable-length character sequences or subword units. BPE is a simple data compression technique that iteratively replaces the most frequent pair of bytes in a sequence with a single, unused byte. In the case of Sennrich's method, instead of merging frequent pairs of bytes, they merged characters or character sequences. Another approach which achieved an open-vocabulary is by Luong et al.[23] used the approach of Sutskever[20] and another LSTM to learn the translation of source word to unknown target word character-by-character.

All of the previous proposed NMT models mentioned produced BLEU scores from 20 to 40 but it all used a parallel corpora composed of around 1,000,000 sentence pairs. A data composed of about 120,000 English-Filipino sentence pairs might not be enough for the NMT to learn the translation. The phenomenon, when a DL model cannot learn well is called underfitting[10]. Another problem that can arise with low data is overfitting. The NMT will be too familiar with the trained data which results to it making large errors when it evaluates the unseen or test data[10]. A common solution for overfitting is dropout[24]. Dropout is a method where a neural network drops some values learned in select hidden units. Another way to solve the problem of low data is using the approach of Zhang et al. [25]. It consists of two neural networks that shares the same encoder but has two different decoders one for machine translation and another for sentence reordering. The machine translation decoder performs the task of the proposed model of Bahdanau [11] using a parallel corpora while the sentence reordering decoder performs the task of reordering a source-side monolingual corpora, the target output is a reordered source sentence. The sentence reordering decoder approximates the target language order by learning the pre-ordering rules proposed by Wang et al. [26]. The approach alternates the two tasks until no im-

provement is seen. First, it performs the sentence reordering task. Afterwards, the encoder parameters learned from the reordering task is used by the machine translation task and so on. The approach gained a 3.45 to 5 BLEU and 1.14 to 1.87 BLEU on 4 different tests performed on a low and high resource pair of the same language pair respectively. Another approach by Sennrich et al.[27] used the monolingual corpora to generate synthetic data by pairing the target monolingual sentences with an empty source sentence or its back-translated sentence. A back-translation is a mechanism that automatically translates the target monolingual sentence into the source language. The back-translation is performed by a baseline NMT. The approach gained 2.1 to 3.4 BLEU on low resource pairs and 2.8 to 3.7 BLEU on high resource pairs and reduced overfitting.

# III. Theoretical Framework

## A. English to Filipino Bidirectional Translations

English to Filipino translations and vice-versa can be done by a human translator, by using a dictionary, or by a machine. English to Filipino bidirectional machine translations is used for communications among nations especially tourists[1] and locals. The goal is to provide high quality translations that is fast and automated.

## B. Neural Machine Translation

Neural machine translation is a recently proposed approach in machine translation. Unlike the traditional statistical machine translation, neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance[11]. The neural networks that can be used for this are Recurrent Neural Networks(RNN), Long Short-Term Memory(LSTM), or Gated Recurrent Units(GRU).

### B..1 Sentence Alignment

This project uses the proposed architecture of Bahdanau et al.[11] or RNNSearch which learns sentence alignments and performs the translation based on the alignment. The set of $h$ is a bidirectional GRU that learns a hypothesis per word $x_t$. The symbol $a$ represents the weights of each hypothesis. The symbol + represents a feed-forward neural network which decides what annotation it passes to $s_t$. The translated word $y_t$ is emitted by $s_t$.

### B..2 Unknown Word Alleviation

This project uses the method proposed by Sennrich et al.[22] that achieved an open-vocabulary translation by encoding rare and unknown words as sequences of subword

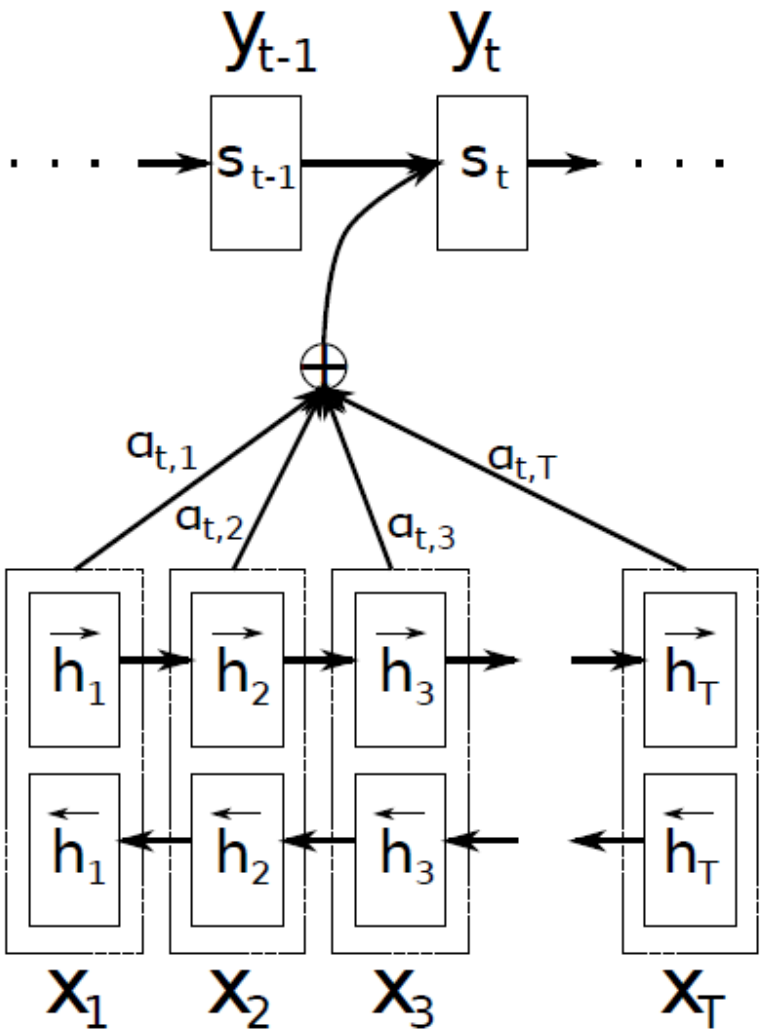


Figure 4: RNN Search model

units.

## C. Recurrent Neural Networks

Recurrent neural networks are a family of neural networks for processing sequential data like sentences[28]. The image used was extracted from colah's blog[29].

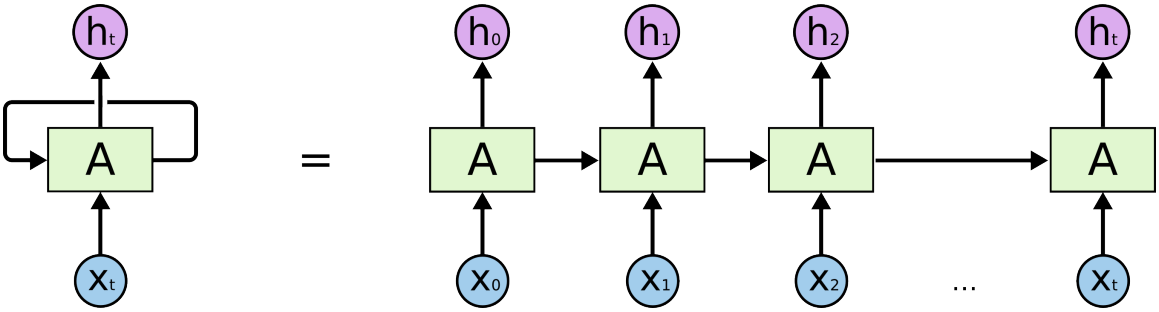


Figure 5: An unrolled model of the RNN

### C..1 Long Short-Term Memory

Are RNN's with memory[10]. It solved the problem of RNN with long-term dependencies [10]. Long-term dependencies are the far relations of inputs in a given sequence. In a sentence, relations like these are 2 or more words apart.

### C..2 Gated Recurrent Units

A simpler model of LSTM. It is much easier to compute and implement[9]. It has two gates, an update gate which controls what information will be updated and a reset gate which controls what information will be removed. The project uses this as a part of the NMT.

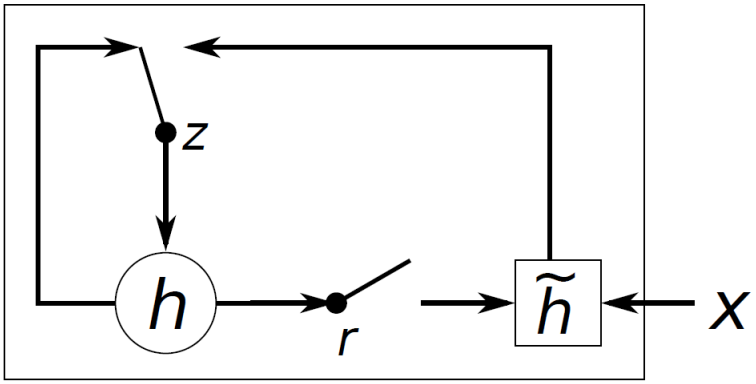


Figure 6: Gated Recurrent Units mechanism

## D. Trainer

A website that utilizes Nematus and produces an NMT model. Additional, the trainer allows an AI Expert to easily deploy a trained model to the web service.

### D..1 Training Page

This is where the AI Expert can train the model. It displays relevant information related to the progress of training.

### D..2 Testing Page

This is where the AI Expert can test the trained model. It displays translations and the BLEU score after testing.

### D..3 Dataset Page

This is where the Language Expert can manage the dataset. It allows the language expert to save suggestions from the Web Service and update the dataset as needed.

## E. Web Service

### E..1 Framework that handles translations and suggestions

A framework that handles post and get requests of web and mobile applications that uses its service. The request translation request is a post request that requests the web service to translate an input using a certain model for that translation. The get translation request is a get request that gets the translation at a certain location. It intuitively provides the status of the translation request. The suggest translation request is a post request that sends a suggestion to the web service. The web service saves the request if its valid. Flask web framework is used for this function.

### E..2 Framework that utilizes Theano

A high-level framework that can easily train, test, and select models. Nematus[30] is used for this function.

### E..3 Theano

A framework for deep learning which has libraries for RNN's and methods that it uses[31].

### E..4 Data Collection

To achieve a large parallel corpora, the proposed system accepts translation suggestions which will be validated by a language expert. Upon reaching 1,000,000 or more sentence pairs, a new model can be retrained for improvements. Redis' database is used for this function.

## F. Mobile Application

An Android-based mobile application which utilizes the web service that provides translations and accepts translation corrections. The web service also supports get and post requests from iOS applications.

## G. Web Application

A Flask-based web application which utilizes the web service that provides translations and accepts translation corrections.

# IV. Design and Implementation

## A. Use Case Diagram

The system has four users the general user, the developer, the language expert, and the AI expert. The general user is the one who will use the application that uses the web service. The developer is the one that will create an application that uses the web service. The language expert will be the one to update or remove datasets. The AI expert will be the one that can perform training and testing of the models used by the system. See the use case diagrams below:

### A..1 Web Service

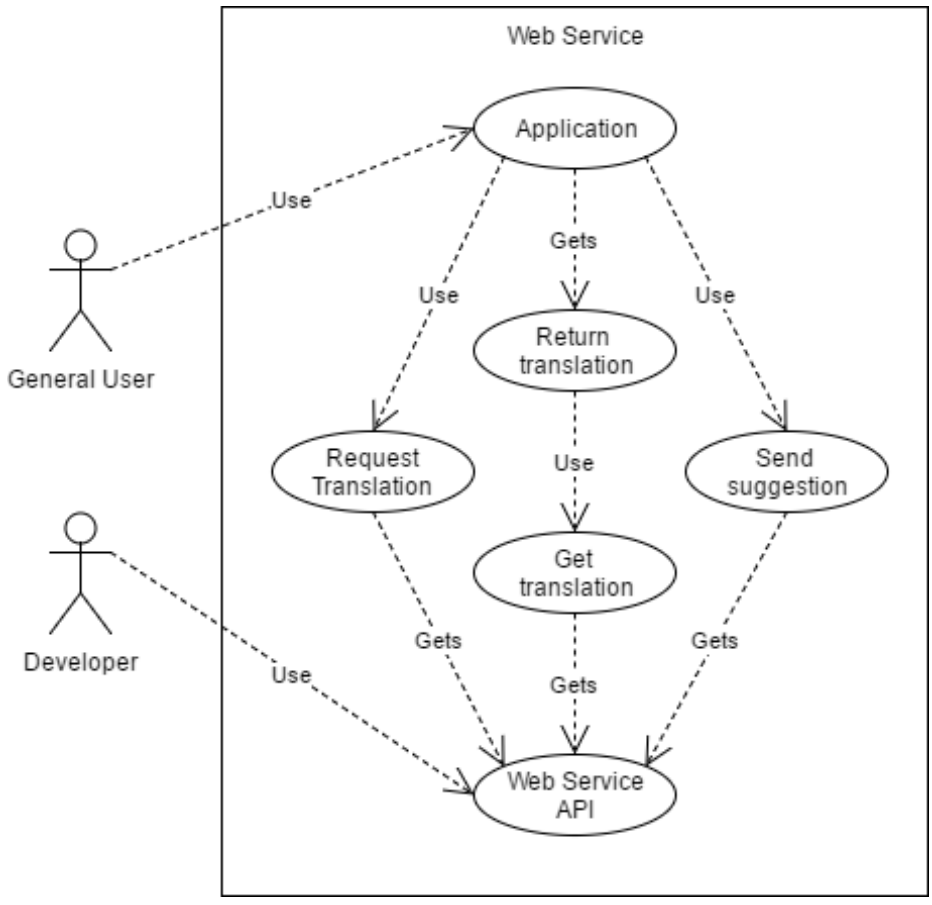


Figure 7: The Web Service

### A..2 Trainer

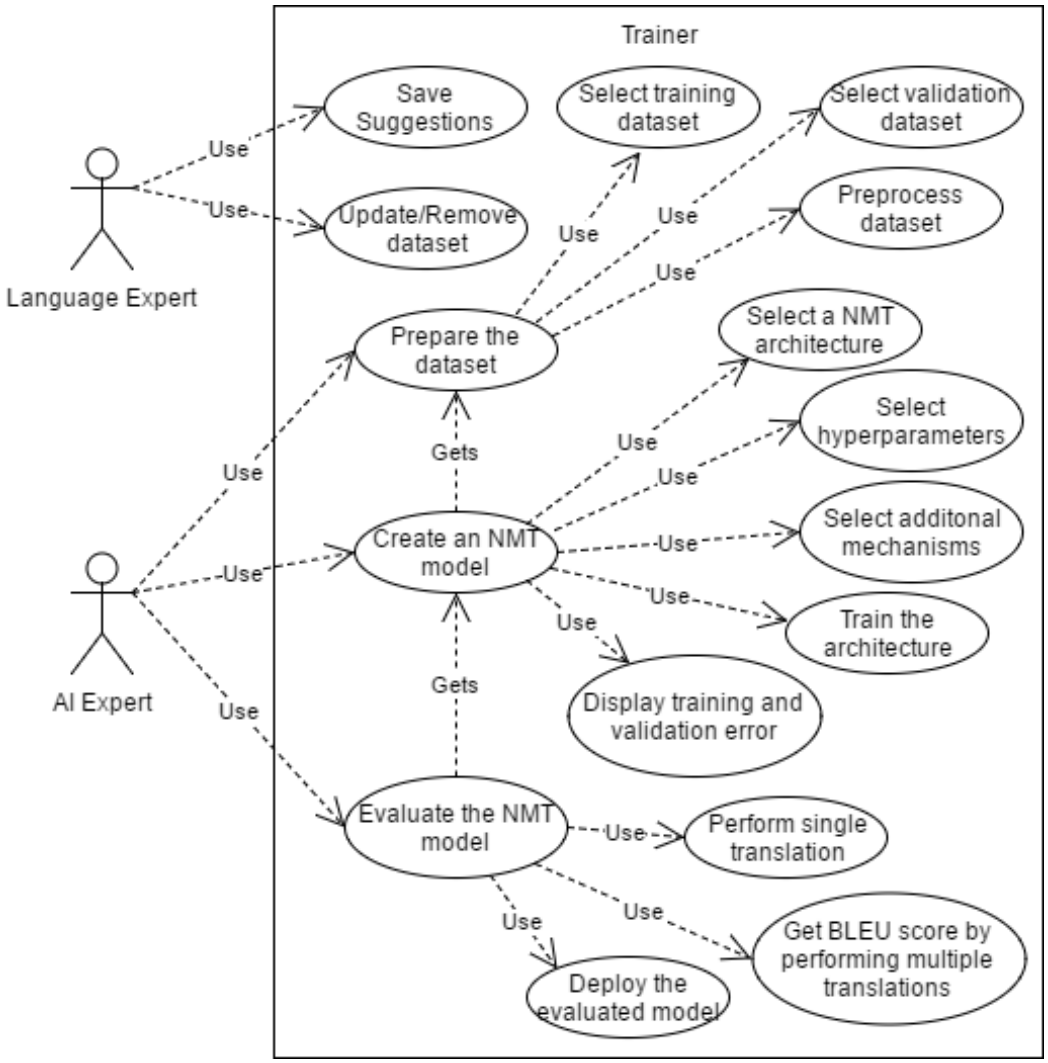


Figure 8: The Trainer

## B. Flow Charts

### B..1 General User

The general user who uses a web or mobile application that integrated the web service. He/She can perform bidirectional Filipino-English translations and also suggest translation corrections.

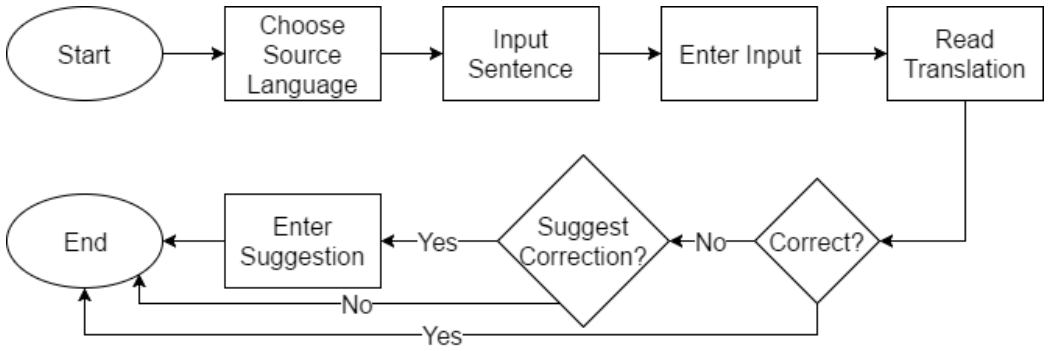


Figure 9: Flow Chart of General User

### B..2 Developer

The developer uses the Web Service Application Program Interface(API) to request and get translations. He/She can also send translation suggestions of the general user.

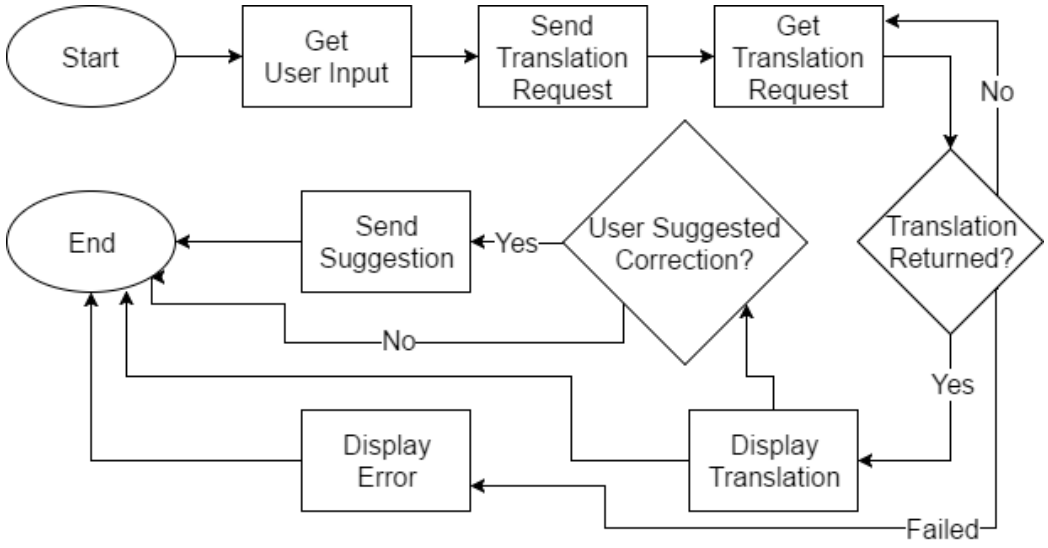


Figure 10: Flow Chart of Developer

### B..3 Language Expert

The suggested translations from the general user can be saved by the language expert. He/She can then update or delete the datasets as needed.

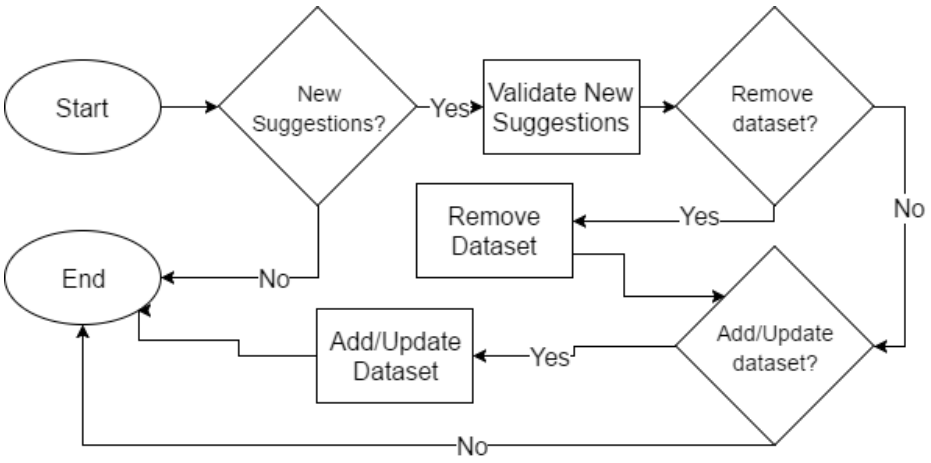


Figure 11: Flow Chart of Language Expert

### B..4 AI Expert

The datasets will be used by the AI Expert to produce better NMT models. If there's a better architecture for the translation task, the AI Expert can create a new model that uses the architecture which can produce better translations. He/She can also evaluate the models produced. If a new model is better than the current deployed model, he/she can deploy the new model.

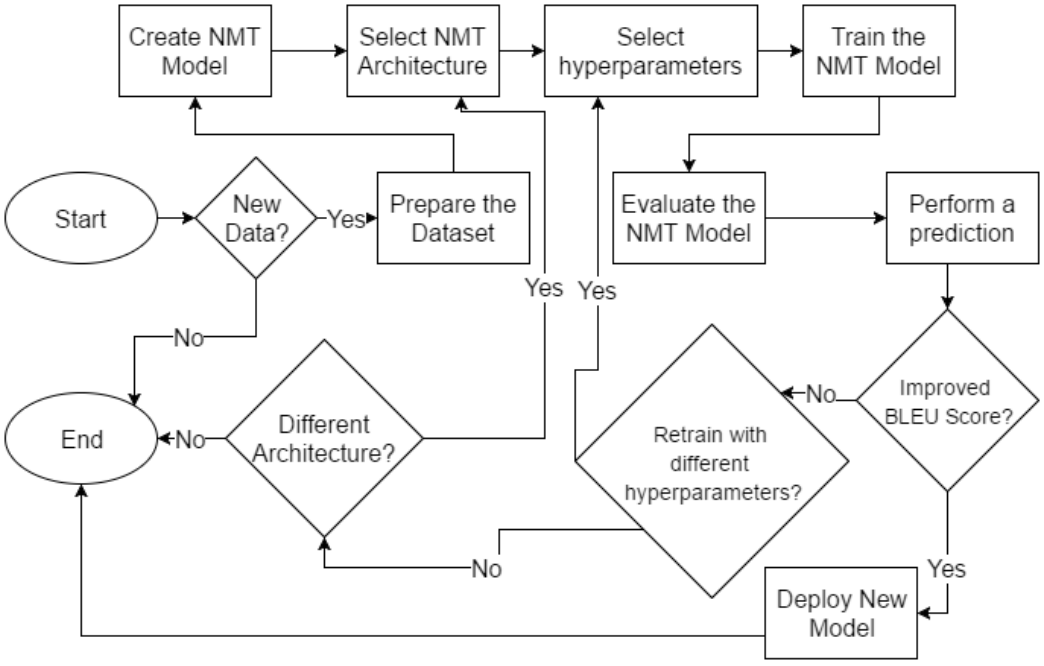


Figure 12: Flow Chart of AI Expert

## C. Technical Architecture

The predicting method that uses a NMT model is implemented in Python using an NMT framework called Nematus[30] and its other dependencies. The Web Service that uses the deployed NMT models is implemented also in Python and uses Flask and Redis.

### C..1 Hardware requirements:

1. CPU rate of 2 GHz or higher
2. Up to 2 GB of free disk space
3. 4 GB of RAM or higher
4. GPU of GTX 750 or better

### C..2 Software requirements:

1. Windows 10 or Ubuntu 16.04
2. Android 4.1 or higher
3. Java JDK 8
4. Python 2.7
5. Theano v0.8.2 (Python)
6. Nematus v0.1.dev0 (Python)
7. Flask v0.12 (Python)
8. Flask-RESTful v0.3.5 (Python)
9. Flask-Bootstrap v3.3.7.1 (Python)

10. Pygal v2.3.1 (Python)
11. Redis v2.10.5 (Python)
12. Celery v3.1.25 (Python)
13. psutil (Python)
14. Cygwin (Windows only)
15. bc (Cygwin)
16. cat (Cygwin)
17. perl (Cygwin and Linux)
18. CUDA v7.5 or v8.0
19. Visual Studio Community 2013 update 5 (Windows only)
20. Redis v3.0

# V. Results

## A. Translation

### A..1 BLEU Score

Our original configuration beats the score of ASEAN MT and Nocon et al. in Filipino-English translations. But using the original configuration in English-Filipino did not even beat the score of Nocon et al. Eventually, the configuration of the 10th experiment created a English-Filipino model that beats Nocon et al. However, the 10th experiment's configuration was not used to create a Filipino-English model due to lack of time, and because it already beats ASEAN MT's BLEU score. The performance of the best Neuro MT models was also tested in a completely different test set, a test set for the tourism domain. The results of the tourism test were projected in Table 2.

Table 1: Test Set BLEU Scores

| | Eng-Fil | Fil-Eng |
|---|---|---|
| Neuro MT(Original) | 29.20 | 37.01* |
| Neuro MT(10th) | 35.25 | – |
| ASEAN MT | 38.12* | 35.79 |
| Nocon et al. | 31.15 | 32.71 |

Table 2: Performance of Best Models

| | Wikipedia | Tourism |
|---|---|---|
| Neuro MT Fil-Eng | 37.01 | 14.86 |
| Neuro MT Eng-Fil | 35.25 | 17.44 |

### A..2 Byte Pair Encoding

The NMT model fails to translate words or subwords existing in the dictionary but not in the data. For words existing in the dictionary or is existing in the data, there are cases when it produces the correct translation.

An example of this is the translation of “pagparami”. The word “pagparami” is not existing in the data and the dictionary. But the subword “pag@@” and the word “parami” are both existing in the dictionary. The translation of “pag” is “.” and the translation of “parami” is “much” while the translation of “pagparami” is “increasing” which is correct.

Another case is when the translation is correct but the subword is division is incorrect. The word “pinakasangkot” was translated as “Most often involved” which is correct. But the division of the word is “pinakas@@” and “angkot”.

Another case is when a word is divided correctly but leads to an incorrect translation. The word “ibigin” is not in the data, it is divided into “ibig@@” and “in”. The word “ibig” is translated as “love” while the word “in” is translated as “in”, the resulting translation is “in”.

These shows that if the division is correct it is up to the subword units if it will lead to a correct translation. In the case of the translation of “pinakasangkot”, the model learned that “pinakas@@” means “the most” and “angkot” means “involved”, this mistake in division will lead to wrong translations when the model is tested on different domains.

### A..3 Sample Translations

This section provides a sample translation of the the best Neuro MT models, ASEAN MT, and Google Translate in our Test Set, in the Tourism Test Set, and in a Wikipedia sample sentence. The Wikipedia sample sentence was extracted from Wikipedia’s new articles and by no means included in our Training set. To see the sample translations, refer to Tables 3 and 4.

Table 3: Sample Translations Eng-Fil

| Test Set | |
|---|---|
| Input | That year Columbus reached the Americas,<br>the beginnings of a global empire. |
| Target | Sa taong iyon, naabot ni Columbus ang Americas,<br>ang simula ng isang pandaigdigang imperyo. |
| Google Translate | Na taon Columbus naabot ang Americas,<br>ang beginnings ng isang pandaigdigang imperyo. |
| ASEAN MT | Iyan ang americas reached columbus taon,<br>ang beginnings ng isang global empire. |
| Neuro MT | Sa taong iyon naabot ng Columbus ang Americas,<br>ang pagsisimula ng pandaigdigang emperyo. |
| Tourism Test Set | |
| Input | I'd like to reserve a table. |
| Target | Gusto kong magpareserba ng mesa. |
| Google Translate | Gusto kong magreserba ng lamesa. |
| ASEAN MT | Gusto i'd magreserba ng table. |
| Neuro MT | Gusto kong reserba ang isang mesa. |
| Wikipedia Sample | |
| Input | Rodrigo "Rody" Roa Duterte, also known as Digong,<br>is a Filipino politician and jurist who is the<br>16th and current President of the Philippines. |
| Target | Si Rodrigo "Rody" Roa Duterte, kilala rin sa kanyang<br>bansag na Digong, ay isang Pilipinong abogado at pulitiko na<br>kasalakuyang naninilbihan bilang ika-16 na Pangulo ng Pilipinas. |
| Google Translate | Rodrigo "Rody" Roa Duterte, na kilala rin<br>bilang Digong, ay isang Pilipinong politiko at dalubhasa sa batas<br>kung sino ang ika-16 at kasalukuyang Presidente ng Pilipinas. |
| ASEAN MT | Duterte, rin rodrigo dahil roa rody digong, ang isang<br>filipino known politician at jurist sino ang 16th at<br>kasalukuyang president ng philippines. |
| Neuro MT | Isang Pilipinong pulitiko at jurist si Rodrigo<br>"Rody" Roa Dlate, na kilala rin bilang Digong,<br>na ika-16 at kasalukuyang Presidente ng Pilipinas. |

Table 4: Sample Translations Fil-Eng

| Test Set | |
|---|---|
| Input | Noong unang bahagi ng Middle Age,<br>napasailalim ito sa Germanic rule. |
| Target | During the early Middle Age<br>it came under Germanic rule. |
| Google Translate | In the early Middle Age,<br>subjected to Germanic rule. |
| ASEAN MT | The first part of the middle age,<br>napasailalim it on germanic rule. |
| Neuro MT | In the early Middle Age,<br>this is subject to Germanic rule. |
| Tourism Test Set | |
| Input | Maaari mo bang ituro ang daan sa pinakamalapit na ospital? |
| Target | Can you show the way to the nearest hospital? |
| Google Translate | Can you point the way to the nearest hospital? |
| ASEAN MT | Can you lead the way to the nearest ospital? |
| Neuro MT | You can tell the road in the closest hospital? |
| Wikipedia Sample | |
| Input | Ang sining ng mga gawaing-kamay ay isang uri<br>ng gawain na ginagamit ang mga mapapakinabangan at<br>madekorasyon na kagamitan sa pamamagitan ng mga kamay o<br>sa paggamit ng mga payak na kagamitan. |
| Target | A handicraft is any of a wide variety<br>of types of work where useful<br>and decorative objects are made completely by hand or<br>by using only simple tools. |
| Google Translate | The handicraft is a kind<br>of work that used the profit and madekorasyon<br>equipment by hand or using simple equipment. |
| ASEAN MT | The art of these gawaing-kamay ay a kind of activity<br>to the use these mapapakinabangan<br>and madekorasyon equipped through hands<br>or in using of these payak : kagamitan. |
| Neuro MT | The art of the hands is a kind<br>of work that would be raised and dramatic<br>tools by hands or by using basic tools. |

# B. REST API

The REST API allows users of the network where the app is deployed to request and suggest translations.

## B..1 API Page

This page shows how to request and suggest translations.

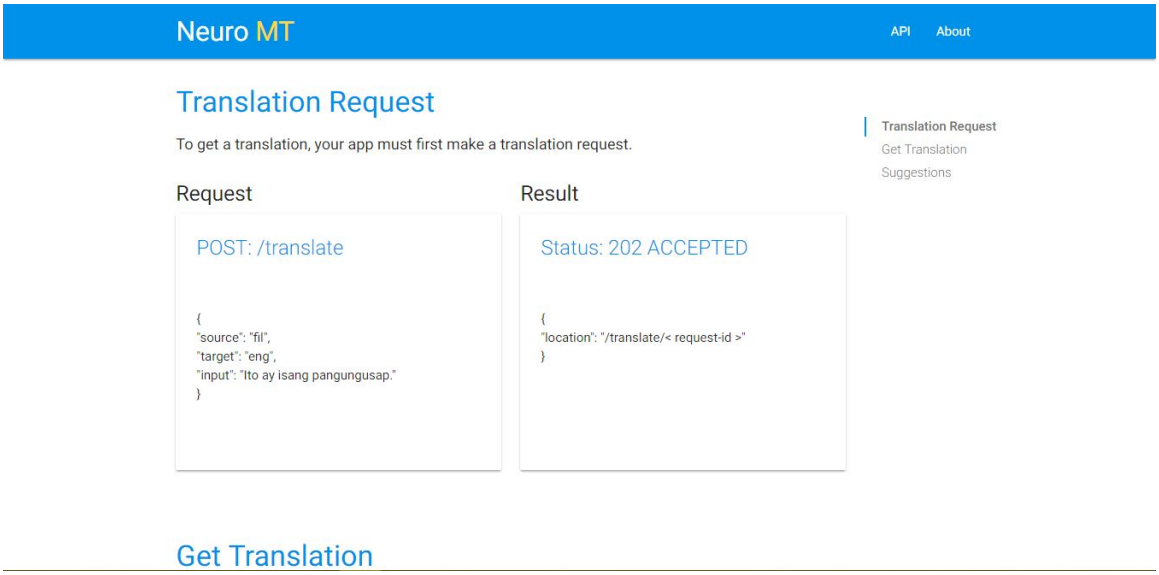


Figure 13: API Page

## B..2 Web App

This web application uses the API. By providing input and clicking the translate button, it can perform translations. By clicking the suggest button after a translation is returned, it can send suggestions.

## B..3 Mobile App

This mobile application uses the API. By providing input and clicking enter, it can perform translations. By clicking the suggest button after a translation is returned, it can send suggestions.

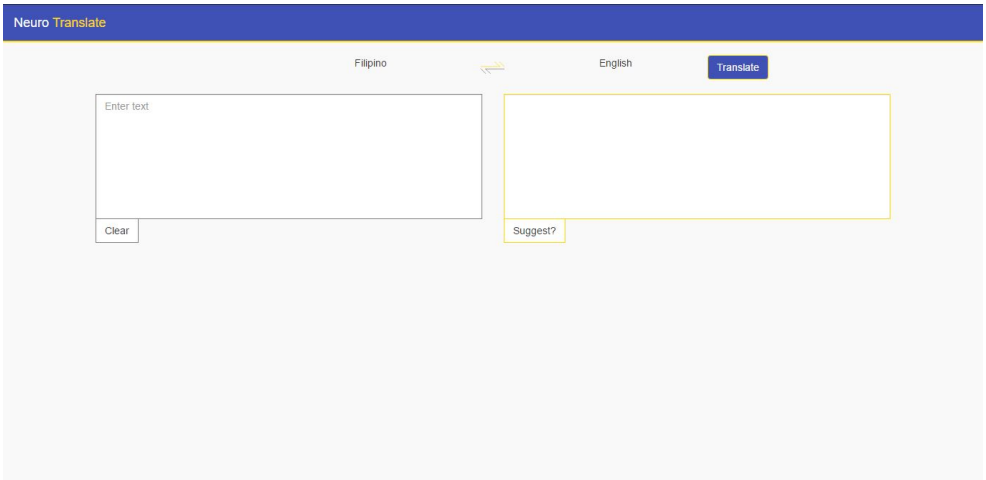


Figure 14: Python Web App

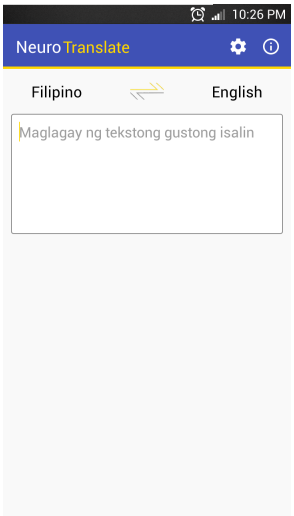


Figure 15: Android Mobile App

## C. Trainer

The Trainer allows an AI Expert and Language Expert to manage the training, testing, and dataset management scenarios.

### C..1 Training

The training page contains a step-by-step instructions and actions before an AI Expert can start training.

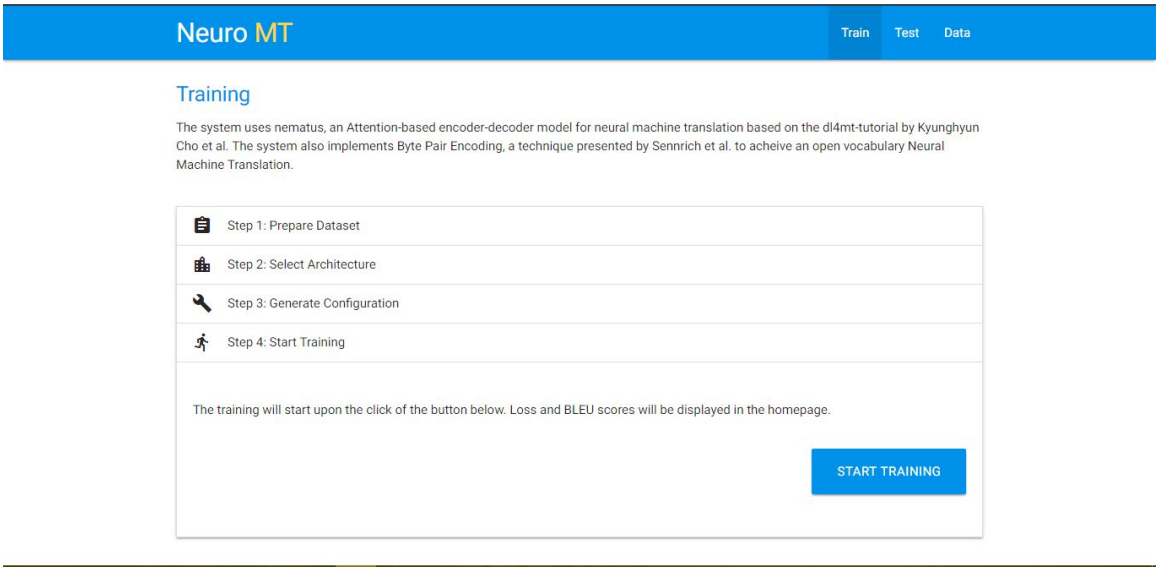


Figure 16: Training Page

## C..2 Testing

The testing page allows an AI Expert to select a model. Upon selecting a model, the AI Expert can perform translations using the model. The AI Expert can also test the model with a test set. And finally, the AI Expert can deploy the model that will be used by the REST API.

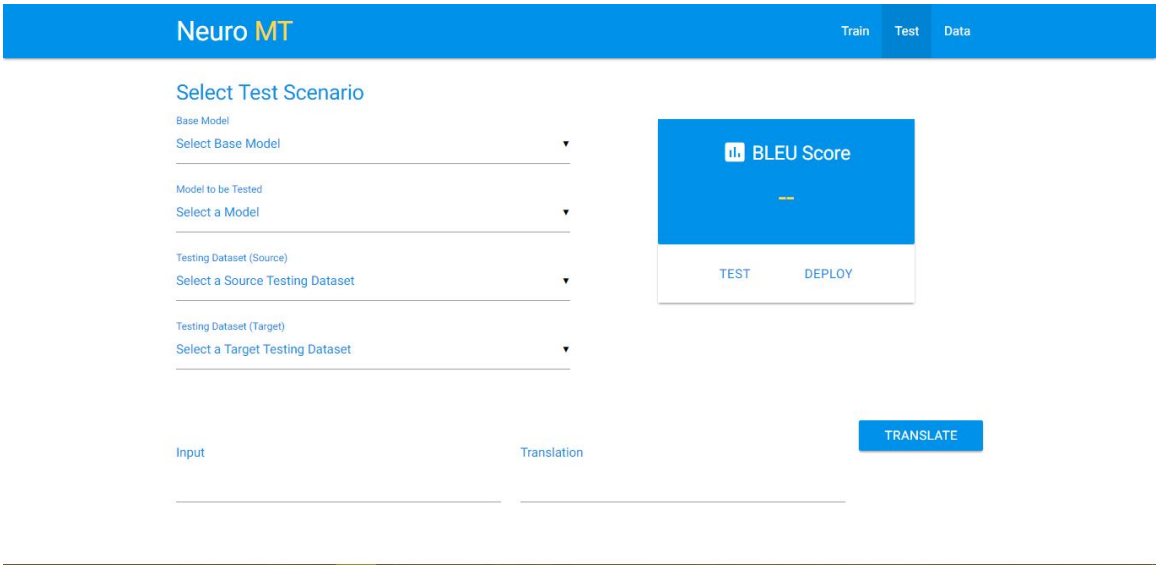


Figure 17: Testing Page

### C..3 Dataset Management

The dataset management page allows a Language Expert to save suggestions, update datasets, and remove datasets.

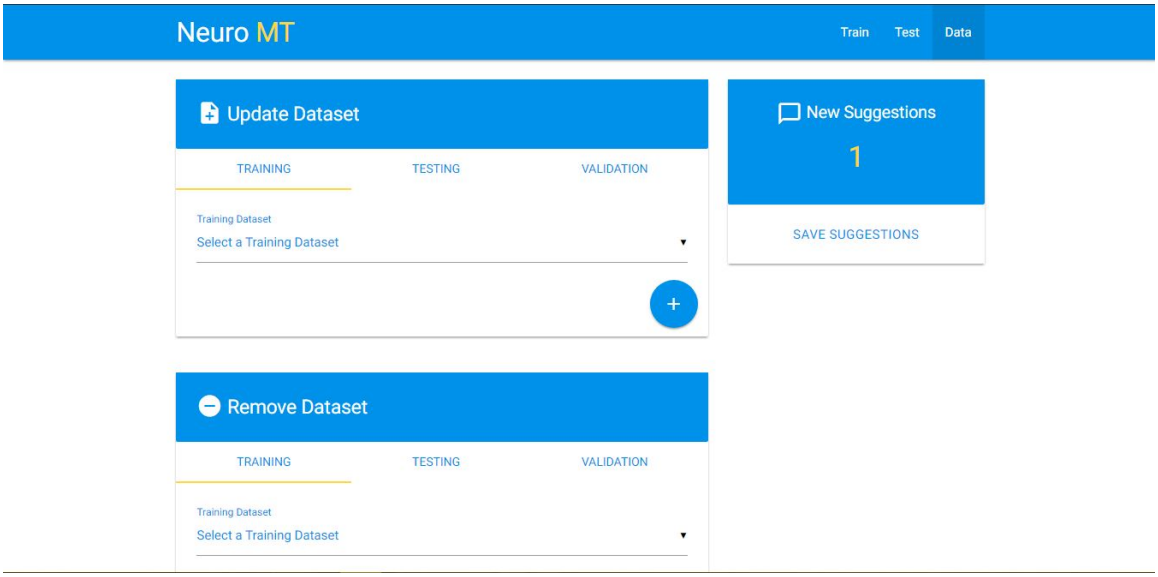


Figure 18: Dataset Management Page

### C..4 Training Progress

The Training progress page has a button that allows an AI Expert to stop the current training. It also displays the training loss, validation loss and the BLEU scores of the whole training.

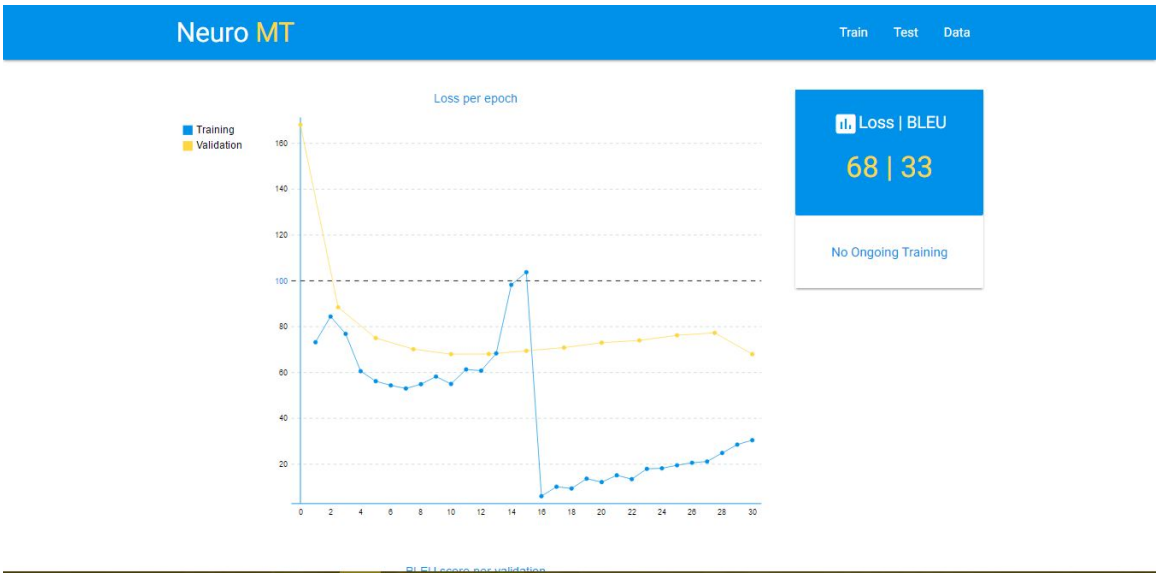


Figure 19: Training Progress Training and Validation Loss

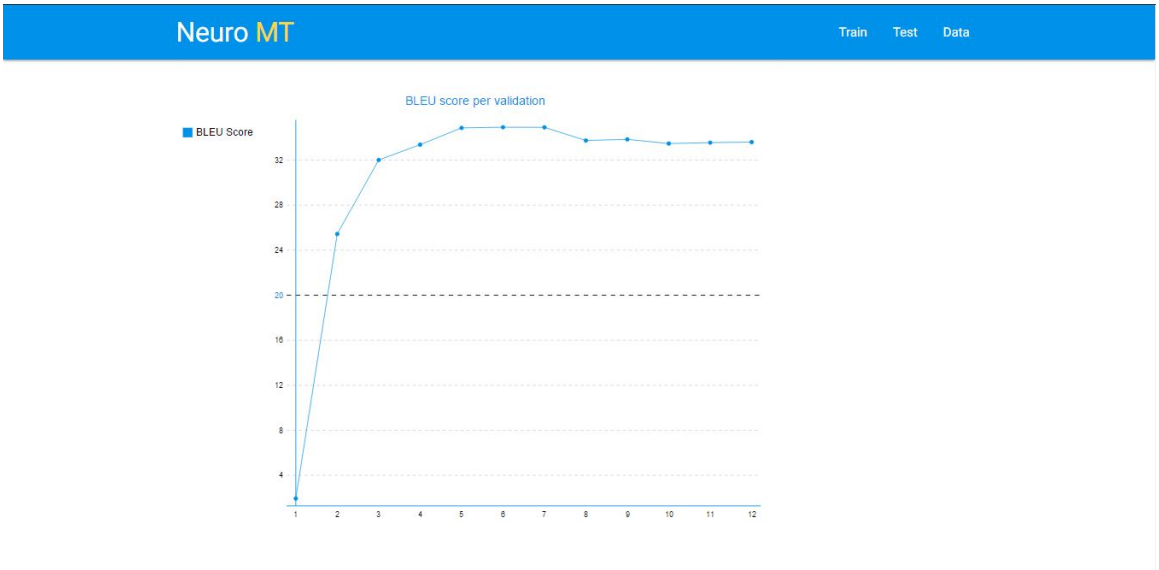


Figure 20: Training Progress BLEU Score

# VI. Discussions

## A. Trainer

### A..1 Objectives

All objectives were fulfilled. Additionally, the "Deploy the evaluated model" objective was added. It is because its logical to put a deploy function if the models are accessed by the web service. This allows the Language Expert to save the suggestions and update the dataset. The dataset allows the AI Expert to improve the models, by learning the new data suggested by the users.

### A..2 Challenges

To support Windows, the older version of some python libraries was used. Additionally, the system does not use the latest version of Nematus, since Nematus is being updated weekly and we can't afford to retrain the models if some incompatibilities occur. The following are the ones affected by these decisions:

1. Flask web framework was used for the web service and trainer because it is lightweight and has a wide array of extensions that supports various tasks. The downside of using Flask is that it is a Synchronous Web Service. It means that for the program to perform other functions, it must first wait for the current task to finish. This resulted to some parts of the system being coded to specifically handle scenarios produced by that design.

2. Celery was used for processing asynchronous requests in the task queue. Version 3.1.25 is the latest version that supports Windows, it became a part of the software requirements. We did not consider making the latest version of Celery work, since it only adds additional features that eases the use of Celery, therefore we do not see any problems if we use an older version.

3. Due to supporting Windows, the only way to stop training in Celery is to prematurely kill the Celery Worker. Killing the Celery Worker means that it clears all the queued tasks to the Worker. But since, all tasks in the Trainer are designed to be performed one at a time, there should be no conflicts if the Celery Worker is prematurely killed.

4. The dictionary of an NMT is where the model gets its representation of words. Losing the dictionary means making the model useless. Since Nematus configures the model to look for its dictionary on a certain position, moving the model's directory may prove difficult. Therefore, we made a folder named "dictionaries", so that even if the models were moved by the system, it will still be able to find its dictionary.

5. Distinguishing models may prove difficult as the number of your trained models increases. Requiring the AI Expert to name the training solved that challenge but required to recode some of the functions to consider the unique name of the models.

6. When to evaluating bad translations, the original code for BLEU score does not return any score. This was solved by manually returning "0.00" BLEU score to the Trainer.

# B. Web Service

## B..1 Objectives

All Objectives were fulfilled. Additionally, an API page was added to the website of the web service. Since it is logical to have a reference of the possible commands that a developer will access for his/her app. This implementation allowed our web and Android application to integrate bidirectional Filipino-English translations.

### B..2 Challenges

1. The usage of Flask means that it is necessary to offload all the translation requests to an asynchronous framework. This means that all the users have to wait for the current translation request to finish before their request can be processed. Luckily, Celery can concurrently process translation requests, for as long as the memory of the computer can handle it. And even if we have used an asynchronous web framework from the start, the problem of limited memory still exists.

2. Distinguishing translation requests from each other. This was solved using the UUID that Celery provides for each task.

3. Integrating a model that is trained with preprocessed data produces a model that requires a preprocessed data. The implementation we have decided upon, is to have the web service call an independent script to preprocess the input sentence and feed it to the model.

4. Not all preprocessing and evaluation scripts are officially ported in Python. The system uses the scripts that Moses SMT uses. Some of its scripts are ported to Python through the nltk package, but the standard evaluation script for BLEU score is inconsistent with its Python implementation. This made us use bash scripts called by Python to get the output of those non-python scripts.

5. Finding a database suitable for translation suggestions and future improvements such as hashing of most frequent translations. Redis is an in-memory data structure store, used as a database, cache and message broker. But it in the early stages of the development it was only used for the message queue to Celery tasks. After researching databases suitable for our system, we found out that Redis has a key-value store. This means that it is designed to handle hashes.

# C. Training Models

## C..1 Objectives

The general objective to use NMT for Filipino to English bidirectional translations was fulfilled. The usage of an NMT allowed a continuous representation of the translation task. Furthermore, it allowed the joint training of its encoder, alignment mechanism, and decoder.

## C..2 Challenges

1. Installation of Nematus and Theano required the installation of its dependencies in both Windows and Ubuntu and Visual Studio Community 2013 update 5 for Windows. The installation of CUDA in Windows is simple but long. The installation of CUDA in Ubuntu requires its drivers to match. But finding the driver that matches the CUDA version is no easy task, since it is undocumented.

2. Installation of Cygwin to allow the Linux-only scripts to run on Windows command line. The challenge in this task is to identify all the Linux commands needed by Nematus, and find its counterpart in Cygwin. Luckily, we were able to find a counterpart for everything, and we were also able to integrate it to the Windows command line by adding it to the environment variables.

3. Producing NMT models rivaling the current state of the art SMT model for Filipino-English bidirectional translations. As seen in the results, only our Filipino to English model was able to beat the state of the art SMT model's BLEU score. Furthermore, it took ten 12-hour trainings to find a configuration that produces an English to Filipino model that beats the score of the 2nd best SMT model.

## D. Mobile App

### D..1 Objectives

All objectives were fulfilled. Additionally, the app records a translation history to take advantage of the queuing mechanism of the REST API.

### D..2 Challenges

1. Since our REST API is portable to any Windows and Linux computers, there's a need to recompile the app to update the location of the server in the network. The original and logical design was to have an uneditable network location at runtime, since a servers do not usually change addresses. The solution we came into was to allow the user to configure the server address, and save it to the SharedPreferences of Android.

## E. Significance of Neuro MT

Since Neuro MT is open-source it can be used by a team of AI Experts, Language Experts, and Developers to

1. Provide semantically and syntactically meaningful bidirectional translations to a target location such as
   (a) Tourist Spots
   (b) Social Research areas
   (c) Schools
2. Accept suggestions from a certain location, which allows mining of location-specific translations
3. Train models based on the dataset formed from a specific location

4. Integrate the web service to web and mobile applications

This project also allows people of other areas to learn a specific translation as they use the system. And finally, the default models of this project can be used as a reference for Neural Machine Translation development in the Philippines.

# VII. Conclusions

Neuro MT is an application with a built-in Web Service and Trainer. The Web Service allows developers and users to perform translations in any app through the use of the models deployed in the Web Service. It also accepts translation suggestions from its users. The Trainer allows the AI Expert to train and test models based on the given dataset, and deploy it to the Web Service afterwards. It also allows a Language Expert to save translation suggestions and update the current dataset.

Neuro MT is developed to provide bidirectional Filipino-English translations to locals, foreigners, researchers, or students. It also allows researchers mine data for location-specific translations, and train Neural Machine Translation(NMT) models based on it.

This project proves that NMT produces semantically and syntactically meaningful representation of translations for bidirectional Filipino-English translations, unlike Statistical Machine Translation(SMT).

The development of this application exposed the limitations of the current research of NMT. We still need to improve the speed of translations, require a lot of GPU Memory, and require a lot of data to learn a translation task, to allow NMT to become more suitable for a web service.

# VIII. Recommendations

## A. Possible System Improvements

The average translation speed when using my Nvidia GeForce GT 635M is 15 seconds. It is possible to improve the Decoder of the NMT by using a different architecture or improving the Beam Search. It is also possible to save the most frequent translations in a HashMap through the use of Redis' database to achieve instant translations. The current implementation of adding language pairs and NMT Frameworks is by adding more conditional statements in the code. To improve this implementation, a function to automatically integrate it to the code can be a possible solution. The overall security of the web service depends only on the built-in security of Flask, Celery, and Redis. It is possible to secure some areas of the web service to prevent attacks. It is also possible to improve the Web Service by allowing developers to cancel translation requests. This removes some unnecessary load acquired by the server through unintended requests. Since the system uses Flask for the Web Service and Trainer, the downside is that it is a Synchronous Web Application. The usage of an Asynchrous Web Application will eliminate the need for Celery and other functions that solely adjusts to the synchronous architecture of Flask. And finally, the translations of the current deployed models are trained using the Wikipedia dataset. This resulted to having formal translations only. It is possible to improve the models by using a conversational dataset. All those suggestions are listed in order of importance:

1. Faster Translations.
2. Cancel Translation Requests.
3. Allow AI or Language Expert to Add or Remove Language Pairs.
4. Allow AI Expert to integrate other NMT Frameworks.

5. Instant Translations.

6. A more secure web service.

7. Improved Conversational Translations.

8. Asynchronous Web Service

## B. Possible Variations

Aside from machine translation, it is possible for an NMT to summarize the a paragraph. Furthermore, it might be possible to translate pseudocodes to a desired language. This can help people who can't code a certain programming language and people who want to simplify complex commands. Basically you can have:

1. News/Sentence Summarization.

2. Pseudocode Translation.

## C. Others

Since the system has a built-in web service, it is possible to have

1. A specialized mobile application

2. A specialized web application

# IX. Bibliography


[1] N. Nocon, N. Oco, J. Ilao, and R. E. Roxas, "Philippine component of the network-based asean language translation public service," in *Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2014 International Conference on*, pp. 1–4, Nov 2014.

[2] W. J. Hutchins, *Machine Translation: Past, Present, Future*. New York, NY, USA: John Wiley & Sons, Inc., 1986.

[3] A. M. TURING, "I.computing machinery and intelligence," *Mind*, vol. LIX, no. 236, pp. 433–460, 1950.

[4] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Dev.*, vol. 3, pp. 210–229, July 1959.

[5] "Statistical machine translation." http://statmt.org/. Accessed: 2016-11-14.

[6] "Cnet." https://www.cnet.com/news/google-translate-now-serves-200-million-people-daily/. Accessed: 2016-11-14.

[7] P. Arthur, G. Neubig, and S. Nakamura, "Incorporating discrete translation lexicons into neural machine translation," in Su *et al.* [32], pp. 1557–1567.

[8] B. Muller, J. Reinhardt, and M. T. Strickland, *Neural networks*. Springer, 1995.

[9] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*

*2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (A. Moschitti, B. Pang, and W. Daelemans, eds.), pp. 1724–1734, ACL, 2014.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[12] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, "Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions," *ArXiv e-prints*, Oct. 2016.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "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, 2002.

[14] "Asean machine translation research." http://www.aseanmt.org/. Accessed: 2016-11-13.

[15] "Watson developer cloud." https://www.ibm.com/watson/developercloud/. Accessed: 2016-11-15.

[16] "Systran translation technologies — online translation, translation software and tools." http://www.systransoft.com/. Accessed: 2016-11-14.

[17] "Google translate api documentation." https://cloud.google.com/translate/docs/. Accessed: 2016-11-14.

[18] J. Ang, M. R. Chan, J. P. Genato, J. Uy, and J. Ilao, "Development of a filipino-to-english bidirectional statistical machine translation system that dynamically updates via user feedback," in *Proceedings of IWSLT*, 2015.

[19] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Baltimore, Maryland), pp. 1370–1380, Association for Computational Linguistics, June 2014.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, 2014.

[21] T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 11–19, The Association for Computer Linguistics, 2015.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers* [33].

[23] M. Luong and C. D. Manning, "Achieving open vocabulary neural machine translation with hybrid word-character models," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers* [33].

[24] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014.

[25] J. Zhang and C. Zong, "Exploiting source-side monolingual data in neural machine translation," in Su *et al.* [32], pp. 1535–1545.

[26] C. Wang, M. Collins, and P. Koehn, "Chinese syntactic reordering for statistical machine translation.," in *EMNLP-CoNLL*, pp. 737–745, 2007.

[27] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers* [33].

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA: MIT Press, 2016.

[29] "Understanding lstm networks." http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 2016-11-14.

[30] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. L"aubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a Toolkit for Neural Machine Translation," in *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, (Valencia, Spain), 2017.

[31] "Welcome theano 0.8.2 documentation." http://deeplearning.net/software/theano/. Accessed: 2016-11-14.

[32] J. Su, X. Carreras, and K. Duh, eds., *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, The Association for Computational Linguistics, 2016.

[33] *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, The Association for Computer Linguistics, 2016.

# X. Appendix

## A. Source Code

### A..1 Neuro MT Codes

File: neuro-mt\oneTranslateGlobal.sh

```
#!/bin/sh

input=$1
model_type=$2
task_id=$3
model_location=$4
nematus=$5

#model_type=$src$tgt

if [ "$model_type" == "engfil" ]
then
    ./translation-scripts/oneTranslate-EngFil.sh "$input"
        $task_id "$model_location" "$nematus"
elif [ "$model_type" == "fileng" ]
then
    ./translation-scripts/oneTranslate-FilEng.sh "$input"
        $task_id "$model_location" "$nematus"
fi
```

File: neuro-mt\postprocess-dev.sh

```
#/bin/sh

# path to moses decoder
moses_scripts=moses-scripts

sed 's/\@\@ //g' | \
perl $moses_scripts/recaser/detruecase.perl
```

File: neuro-mt\preprocess.sh

```
#!/bin/sh

# this sample script preprocesses a sample corpus, including
      tokenization,
# truecasing, and subword segmentation.
# for application to a different language pair,
# change source and target prefix, optionally the number of
      BPE operations,
# and the file names (currently, data/corpus and data/
      newsdev2016 are being processed)

# in the tokenization step, you will want to remove Romanian
      -specific normalization / diacritic removal,
# and you may want to add your own.
# also, you may want to learn BPE segmentations separately for
       each language,
# especially if they differ in their alphabet

# name of training
name=$1

# name of training set
t_name=$2

# name of developement set
d_name=$3

# suffix of source language files
SRC=$4

# suffix of target language files
TRG=$5

# max length of sentences
max_length=$6

# number of merge operations. Network vocabulary should be
      slightly larger (to include characters),
# or smaller if the operations are learned on the joint
      vocabulary
bpe_operations=$7 #49500 #89500 , Vocabulary size - 500

# path to moses decoder: https://github.com/moses-smt/
      mosesdecoder
moses_scripts=moses-scripts

# path to subword segmentation scripts: https://github.com/
      rsennrich/subword-nmt
subword_nmt=subword-nmt-master

# path to nematus ( https://www.github.com/rsennrich/
      nematus )
nematus=$8 #/path/to/nematus

# path to data
static_training_data=static/data/train
static_development_data=static/data/dev
data_folder=data_training
model_folder=model_training
dictionary_folder=dictionaries

# move the selected files to the data_folder
cp $static_training_data/$t_name.$SRC $data_folder/train_data.
      $SRC
cp $static_training_data/$t_name.$TRG $data_folder/train_data.
      $TRG
cp $static_development_data/$d_name.$SRC $data_folder/
      dev_data.$SRC
cp $static_development_data/$d_name.$TRG $data_folder/
      dev_data.$TRG

# tokenize
for prefix in train_data dev_data
 do
   cat $data_folder/$prefix.$SRC | \
   perl $moses_scripts/tokenizer/normalize-punctuation.perl -l
        $SRC | \
   perl $moses_scripts/tokenizer/tokenizer.perl -a -l $SRC >
        $data_folder/$prefix.tok.$SRC

   cat $data_folder/$prefix.$TRG | \
   perl $moses_scripts/tokenizer/normalize-punctuation.perl -l
        $TRG | \
   perl $moses_scripts/tokenizer/tokenizer.perl -a -l $TRG >
        $data_folder/$prefix.tok.$TRG

 done

# clean empty and long sentences, and sentences with high
      source-target ratio (training corpus only)
perl $moses_scripts/training/clean-corpus-n.perl $data_folder/
      train_data.tok $SRC $TRG $data_folder/train_data.tok.
      clean 1 $max_length

# train truecaser
perl $moses_scripts/recaser/train-truecaser.perl -corpus
      $data_folder/train_data.tok.clean.$SRC -model
      $model_folder/truecase-model.$SRC
perl $moses_scripts/recaser/train-truecaser.perl -corpus
      $data_folder/train_data.tok.clean.$TRG -model
      $model_folder/truecase-model.$TRG

# apply truecaser (cleaned training corpus)
for prefix in train_data
 do
  perl $moses_scripts/recaser/truecase.perl -model
       $model_folder/truecase-model.$SRC < $data_folder/
       $prefix.tok.clean.$SRC > $data_folder/$prefix.tc.$SRC
  perl $moses_scripts/recaser/truecase.perl -model
       $model_folder/truecase-model.$TRG < $data_folder/
       $prefix.tok.clean.$TRG > $data_folder/$prefix.tc.$TRG
 done

# apply truecaser (dev files)
for prefix in dev_data
 do
  perl $moses_scripts/recaser/truecase.perl -model
       $model_folder/truecase-model.$SRC < $data_folder/
       $prefix.tok.$SRC > $data_folder/$prefix.tc.$SRC
  perl $moses_scripts/recaser/truecase.perl -model
       $model_folder/truecase-model.$TRG < $data_folder/
       $prefix.tok.$TRG > $data_folder/$prefix.tc.$TRG
 done

echo Training BPE with $bpe_operations merge operations
```

```
# train BPE    (BPE allows an open-dictionary
      implementation, this was proposed by Sennrich et al. See
      paper "NMT of Rare Words with subword units")
cat $data_folder/train_data.tc.$SRC $data_folder/train_data.tc.
      $TRG | python $subword_nmt/learn_bpe.py -s
      $bpe_operations > $model_folder/$SRC$TRG.bpe

# apply BPE

for prefix in train_data dev_data
 do
  python $subword_nmt/apply_bpe.py -c $model_folder/
        $SRC$TRG.bpe < $data_folder/$prefix.tc.$SRC >
        $data_folder/$prefix.bpe.$SRC
  python $subword_nmt/apply_bpe.py -c $model_folder/
        $SRC$TRG.bpe < $data_folder/$prefix.tc.$TRG >
        $data_folder/$prefix.bpe.$TRG
 done

# build network dictionary
python $nematus/data/build_dictionary.py $data_folder/
      train_data.bpe.$SRC $data_folder/train_data.bpe.$TRG

# move dictionary to a general directory
cp $data_folder/train_data.bpe.$SRC.json $dictionary_folder/
      $name.bpe.$SRC.json
cp $data_folder/train_data.bpe.$TRG.json $dictionary_folder/
      $name.bpe.$TRG.json


File: neuro-mt\REST_API.py

import os
import subprocess
import redis
import json
from flask import Flask, request, render_template,
      send_from_directory, \
     url_for, jsonify, current_app
from flask_restful import Resource, Api, reqparse, fields,
      marshal, marshal_with
from gevent.wsgi import WSGIServer
from flask_bootstrap import Bootstrap
from celery import Celery
from werkzeug.debug import DebuggedApplication

app = Flask(__name__)
app.config['SECRET_KEY'] = 'only-the-chosen-ones-know-
      it'

#Celery Configuration
app.config['CELERY_BROKER_URL'] = 'redis://localhost
      :6379/0'
app.config['CELERY_RESULT_BACKEND'] = 'redis://localhost
      :6379/0'
app.config['NEMATUS_PATH'] = 'C:/Users/Alfred/Anaconda2/
      Lib/site-packages/nematus'

celery = Celery(app.name, broker=app.config['
      CELERY_BROKER_URL'])
celery.conf.update(app.config)

Bootstrap(app)

app.config['BOOTSTRAP_SERVE_LOCAL'] = True

db = redis.Redis('localhost')

@app.route('/')
def index():
    return render_template("index_rest.html")

class InvalidTranslationTask(Exception):
    pass

custom_errors = {
    'InvalidTranslationTask': {
        'message': "Source or Target language is invalid.",
        'status': 409,
    }
}

api = Api(app, errors=custom_errors)

suggestion_fields = {
    'eng_data' : fields.List(fields.String),
    'fil_data' : fields.List(fields.String)
}

#STATES: PENDING, FAILURE
class Translation(Resource):
    def get(self):
        pass

    def post(self):
        parser = reqparse.RequestParser()
        parser.add_argument('source', required=True, type=str,
              location='json')
        parser.add_argument('target', required=True, type=str,
              location='json')
        parser.add_argument('input', required=True, type=str,
              location='json')

        args = parser.parse_args(strict=True)
        source = args['source']
        target = args['target']
        input = args['input']

        #print input
        if ((source == 'eng' and target == 'fil') or (source ==
              'fil' and target == 'eng')):
            task = translation_task.apply_async(args=[source,
                  target, input, current_app.config.get('
                  NEMATUS_PATH')])
            return {'location': url_for('translationstatus',
                  task_id=task.id)}, 202
        raise InvalidTranslationTask
api.add_resource(Translation, '/translate')

class TranslationStatus(Resource):
    def get(self, task_id):
        #print task_id
        task = translation_task.AsyncResult(task_id)
        if task.state == 'PENDING':
            response = {
                'state': task.state,
                'translation': ''
            }
        elif task.state != 'FAILURE':
            response = {
                'state': task.state,
                'translation': task.info.get('translation')
            }
        else:
            # something went wrong in the background job
            response = {
                'state': task.state,
                'translation': str(task.info)  # this is the
                      exception raised
            }
        return response #fix with jsonify
api.add_resource(TranslationStatus, '/translate/<task_id>')

class Suggest(Resource):
    def get(self):
        pass

    def post(self):
        parser = reqparse.RequestParser()
        parser.add_argument('eng', required=True, type=str,
              location='json')
        parser.add_argument('fil', required=True, type=str,
              location='json')

        args = parser.parse_args(strict=True)
        english = args['eng']
        filipino = args['fil']

        english = " ".join(english.split()) #remove duplicate
              spaces
        filipino = " ".join(filipino.split())

        english = english.strip() #remove leading and ending
              spaces
        filipino = filipino.strip()

        if (english != "" and filipino != ""):
            db.rpush('english_suggestions', english)
            db.rpush('filipino_suggestions', filipino)

        return {}, 200
api.add_resource(Suggest, '/suggest')

@app.route('/favicon.ico')
def favicon():
    return send_from_directory(os.path.join(app.root_path, '
          static'),
                               'favicon.ico', mimetype='image
                                     /vnd.microsoft.icon')

@app.route('/translateapp', methods = ['GET'])
def translateapp():
    return render_template("translate.html")
```

```
@app.route('/api', methods = ['GET'])
def api():
    return render_template("api.html")

@app.route('/about', methods = ['GET'])
def about():
    return render_template("about.html")

@celery.task(bind=True)
def translation_task(self, source, target, input, nematus):
    print source
    print target
    print input
    self.update_state(state='TRANSLATING', meta={'
        translation': ''})
    print self.request.id
    task_number = self.request.id
    model_fileng = 'model_deployed/fileng.npz'
    model_engfil = 'model_deployed/engfil.npz'
    if(source == 'fil' and target == 'eng'):
        log = subprocess.Popen(['bash', 'oneTranslateGlobal.sh',
            input, 'fileng', task_number, model_fileng,
            nematus], stdout=subprocess.PIPE, stderr=
            subprocess.PIPE)
    elif(source == 'eng' and target == 'fil'):
        log = subprocess.Popen(['bash', 'oneTranslateGlobal.sh',
            input, 'engfil', task_number, model_engfil,
            nematus], stdout=subprocess.PIPE, stderr=
            subprocess.PIPE)
    output, error = log.communicate()
    print output
    print error
    return {'translation': output.strip()}

if __name__ == '__main__':
    server = WSGIServer(('0.0.0.0', 80), DebuggedApplication(
        app))
    server.serve_forever()
```

File: neuro-mt\test.sh

```
#!/bin/sh

# theano device, in case you do not want to compute on gpu,
    change it to cpu
device=gpu

# path to moses scripts
moses_scripts=moses-scripts

# path to subword nmt
subword_nmt=subword-nmt-master

# name of base model
base_model=$1

# name of model
model_name=$2

# name of test set
t_name=$3

# src language
SRC=$4

# target language
TRG=$5

# path to nematus ( https://www.github.com/rsennrich/
    nematus )
nematus=$6 #/path/to/nematus

# max length of sentences
#max_length=$5

# language direction
#model_type=$SRC$TRG

# path to model
model=model_training/$model_name

# path to data
static_testing_data =static/data/test
data_folder=data_testing
model_folder=model_training

# move the selected files to the data_folder
cp $static_testing_data /$t_name.$SRC $data_folder/test_data.
    $SRC
cp $static_testing_data /$t_name.$TRG $data_folder/test_data.
    $TRG

# copy config of base model
cp $model_folder/$base_model.json $model_folder/$model_name.
    json

# tokenize
for prefix in test_data
 do
   cat $data_folder/$prefix.$SRC | \
   perl $moses_scripts/tokenizer/normalize-punctuation.perl -l
       $SRC | \
   perl $moses_scripts/tokenizer/tokenizer.perl -a -l $SRC >
       $data_folder/$prefix.tok.$SRC

   cat $data_folder/$prefix.$TRG | \
   perl $moses_scripts/tokenizer/normalize-punctuation.perl -l
       $TRG | \
   perl $moses_scripts/tokenizer/tokenizer.perl -a -l $TRG >
       $data_folder/$prefix.tok.$TRG

 done

# A truecaser is automatically generated when Preparing Data.
# But there is a best practice when training a truecase model.
# An AI Expert is required to manually replace the truecase-
    model in the truecasers folder before testing.
# apply truecaser
for prefix in test_data
 do
  perl $moses_scripts/recaser/truecase.perl -model truecasers/
      truecase-model.$SRC < $data_folder/$prefix.tok.$SRC
      > $data_folder/$prefix.tc.$SRC
  perl $moses_scripts/recaser/truecase.perl -model truecasers/
      truecase-model.$TRG < $data_folder/$prefix.tok.$TRG
      > $data_folder/$prefix.tc.$TRG
 done

# apply BPE

for prefix in test_data
 do
  python $subword_nmt/apply_bpe.py -c $model_folder/
      $SRC$TRG.bpe < $data_folder/$prefix.tc.$SRC >
      $data_folder/$prefix.bpe.$SRC
  python $subword_nmt/apply_bpe.py -c $model_folder/
      $SRC$TRG.bpe < $data_folder/$prefix.tc.$TRG >
      $data_folder/$prefix.bpe.$TRG
 done

dev=$data_folder/test_data.bpe.$SRC
ref=$data_folder/test_data.tok.$TRG

touch $dev.output.dev

# decode

THEANO_FLAGS=mode=FAST_RUN,floatX=float32,device=
    $device,on_unused_input=warn python $nematus/nematus/
    translate.py \
    -m $model \
    -i $dev \
    -o $dev.output.dev \
    -k 12 -n -p 1

./postprocess-dev.sh < $dev.output.dev > $dev.output.
    postprocessed.dev

## get BLEU
$moses_scripts/generic/multi-bleu.perl $ref < $dev.output.
    postprocessed.dev >> logs/${prefix}_bleu_scores
BLEU=`$moses_scripts/generic/multi-bleu.perl $ref < $dev.
    output.postprocessed.dev | cut -f 3 -d ' ' | cut -f 1 -d
    ','`

echo "$BLEU"

#rm $dev.output.dev
```

File: neuro-mt\train.sh

```
# theano device, in case you do not want to compute on gpu,
    change it to cpu
device=gpu

#THEANO_FLAGS=mode=FAST_RUN,floatX=float32,device=
    $device,on_unused_input=warn python training_config.py
THEANO_FLAGS=optimizer=fast_compile,floatX=float32,nvcc.
    flags=-arch=sm_21,device=$device,on_unused_input=warn
     python training_config.py
```

File: neuro-mt\Trainer.py

```
import os
import time
import io
import subprocess
import sys
import redis
import json
import shutil
import pygal
import psutil
from flask import Flask, request, render_template,
    send_from_directory, \
    url_for, jsonify, current_app
from gevent.wsgi import WSGIServer
from flask_restful import Resource, Api, reqparse
from celery import Celery
from celery.result import AsyncResult
from celery.contrib.abortable import AbortableTask
from celery.contrib.abortable import AbortableAsyncResult
from werkzeug.utils import secure_filename
from werkzeug.debug import DebuggedApplication
from pygal.style import Style

app = Flask(__name__)
app.config['SECRET_KEY'] = 'its-a-secret'

#Celery Configuration
app.config['CELERY_BROKER_URL'] = 'redis://localhost
    :6379/0'
app.config['CELERY_RESULT_BACKEND'] = 'redis://localhost
    :6379/0'
app.config['NEMATUS_PATH'] = 'C:/Users/Alfred/Anaconda2/
    Lib/site-packages/nematus'

celery = Celery(app.name, broker=app.config['
    CELERY_BROKER_URL'])
celery.conf.update(app.config)

db = redis.Redis('localhost') #connect to server
api = Api(app)
trainId = "0"

class PreprocessStatus(Resource):
    def get(self, task_id):
        print task_id
        task = preprocess_task.AsyncResult(task_id)
        response = {
            'state': task.state
        }
        return response
api.add_resource(PreprocessStatus, '/preprocessstatus/<task_id
    >', endpoint="api.preprocessstatus")

class DeployStatus(Resource):
    def get(self, task_id):
        print task_id
        task = deploy_task.AsyncResult(task_id)
        response = {
            'state': task.state
        }
        return response
api.add_resource(DeployStatus, '/deploystatus/<task_id>',
    endpoint="api.deploystatus")


class SaveStatus(Resource):
    def get(self, task_id):
        print task_id
        task = save_task.AsyncResult(task_id)
        response = {
            'state': task.state
        }
        return response
api.add_resource(SaveStatus, '/savestatus/<task_id>', endpoint
    ="api.savestatus")


class UpdateStatus(Resource):
    def get(self, task_id):
        print task_id
        task = update_task.AsyncResult(task_id)
        response = {
            'state': task.state
        }
        return response
api.add_resource(UpdateStatus, '/updatestatus/<task_id>',
    endpoint="api.updatestatus")


class RemoveStatus(Resource):
    def get(self, task_id):
        print task_id
        task = test_task.AsyncResult(task_id)
        response = {
            'state': task.state
        }
        return response
api.add_resource(RemoveStatus, '/removestatus/<task_id>',
    endpoint="api.removestatus")

class TestStatus(Resource):
    def get(self, task_id):
        print task_id
        task = test_task.AsyncResult(task_id)
        if task.state == 'PENDING':
            response = {
                'state': task.state,
                'bleuScore': 'Pending...'
            }
        elif task.state != 'FAILURE':
            response = {
                'state': task.state,
                'bleuScore': task.info.get('bleuScore')
            }
        else:
            # something went wrong in the background job
            response = {
                'state': task.state,
                'bleuScore': str(task.info)  # this is the
                    exception raised
            }
        return response
api.add_resource(TestStatus, '/teststatus/<task_id>', endpoint
    ="api.teststatus")

class TranslationStatus(Resource):
    def get(self, task_id):
        print task_id
        task = translation_task.AsyncResult(task_id)
        if task.state == 'PENDING':
            response = {
                'state': task.state,
                'translation': 'Pending...'
            }
        elif task.state != 'FAILURE':
            response = {
                'state': task.state,
                'translation': task.info.get('translation')
            }
        else:
            # something went wrong in the background job
            response = {
                'state': task.state,
                'translation': str(task.info)  # this is the
                    exception raised
            }
        return response
api.add_resource(TranslationStatus, '/translationstatus/<
    task_id>', endpoint="api.translationstatus")

@app.route('/')
def index():
    custom_style = Style(
      background='#ffffff',
      plot_background='#ffffff',
      foreground='#000000',
      foreground_strong='#0091ea',
      foreground_subtle='#cfd8dc',
      opacity='1',
      opacity_hover='0.9',
      transition='400ms ease-in',
      font_family='sans-serif',
      colors=('#0091ea', '#ffd740'))

    global trainId

    state = train_task.AsyncResult(trainId).state #states:
        PENDING, TRAINING, ABORTED/FAILURE
    validFreq = 2.5
    dispFreq = 1
    trainName = ""
    train_array = []
    valid_array = []
    bleu_array = []
    bleu_label_array = []
    train_counter = 1
    valid_counter = 0
    bleu_counter = 1

    if(os.path.isfile("logs/train.log")):
        with open("logs/train.log", "r") as f:
            for line in f:
                words = line.split(" ")
```

```
                if words[0] == "Name":
                    trainName = words[1]
                    dispFreq = int(words[3])
                    validFreq = int(words[5].rstrip('\n'))
                elif words[0] == "Valid":
                    valid = float(words[2].rstrip('\n'))
                    valid_array.append((valid_counter *
                        validFreq, valid))
                    valid_counter += 1
                elif words[0] == "Epoch":
                    cost = float(words[8].rstrip('\n'))
                    train_array.append((train_counter *
                        dispFreq, cost))
                    train_counter += 1
                elif words[0] == "BLEU":
                    bleu = float(words[2].rstrip('\n'))
                    bleu_array.append(bleu)
                    bleu_label_array.append(bleu_counter)
                    bleu_counter += 1
        f.close;
    else:
        trainName = "No Ongoing Training"

    config = pygal.Config(style=custom_style)
    #adds tooltips of pygal which is only available when the
        server is online
    config.js = ['http://localhost:5000' + url_for('.static',
        filename='pygal-tooltips.js')]
    graph = pygal.XY(config)
    graph.title = 'Loss per epoch' #should be updated by
        prepare
    graph.add('Training', train_array)
    graph.add('Validation', valid_array)
    graph_data = graph.render_data_uri()

    graph2 = pygal.Line(config)
    graph2.title = 'BLEU score per validation'
    graph2.x_labels = bleu_label_array
    graph2.add('BLEU Score', bleu_array)
    graph_data2 = graph2.render_data_uri()

    bleu = "--"
    loss = "--"

    if bleu_array:
        bleu = int(bleu_array[-1])

    if valid_array:
        loss = int(valid_array[-1][1])

    return render_template("index.html", graph_data =
        graph_data, graph_data2 = graph_data2, bleu=bleu,
        loss=loss, trainName=trainName, state=state)

@app.route('/train')
def train():
    training_list = os.listdir(os.path.join(app.static_folder, '
        data/train'))
    development_list = os.listdir(os.path.join(app.static_folder
        , 'data/dev'))
    return render_template("train.html", tlist = training_list,
        dlist = development_list)

@app.route('/getFileLength', methods = ['POST'])
def getFileLength():
    type = request.form['type']
    name = request.form['name']
    count = 0
    if type == "train" and os.path.isfile(os.path.join(app.
        static_folder, 'data/train', name)):
        count = file_len(os.path.join(app.static_folder, 'data/
            train', name))
    elif type == "dev" and os.path.isfile(os.path.join(app.
        static_folder, 'data/dev', name)):
        count = file_len(os.path.join(app.static_folder, 'data/
            dev', name))

    return jsonify({'count': count}), 200

@app.route('/test')
def test():
    test_list = os.listdir(os.path.join(app.static_folder, '
        data/test'))
    model_list = []
    for file in os.listdir(os.path.join(app.root_path, '
        model_training')):
        if (file.endswith(".npz")):
            model_list.append(file)
    return render_template("test.html", tlist = test_list, mlist
        = model_list)

@app.route('/data')
def data():
    training_list = os.listdir(os.path.join(app.static_folder, '
        data/train'))
    testing_list = os.listdir(os.path.join(app.static_folder, '
        data/test'))
    development_list = os.listdir(os.path.join(app.static_folder
        , 'data/dev'))
    new_data = os.listdir(os.path.join(app.root_path, 'new data
        '))
    size = db.llen("english_suggestions")
    return render_template("data.html", suggestions_count =
        size, tnlist = training_list, ttlist = testing_list,
        dlist = development_list, ndlist = new_data)

@app.route('/data_save')
def data_save():
    task_save = save_task.apply_async()
    return jsonify({'location': url_for('api.savestatus',
        task_id=task_save.id)}), 202

@celery.task(bind=True)
def save_task(self):
    self.update_state(state='SAVING')
    #generate suggestions file
    eng = db.lrange("english_suggestions", 0, -1)
    fil = db.lrange("filipino_suggestions", 0, -1)

    f = open("suggestions/suggestions.eng","w") #opens file
        with name of "test.txt"
    for sentence in eng:
        f.write(sentence + "\n")
    f.close()

    f = open("suggestions/suggestions.fil","w") #opens file
        with name of "test.txt"
    for sentence in fil:
        f.write(sentence + "\n")
    f.close()

    db.delete("english_suggestions")
    db.delete("filipino_suggestions")
    return {}

@app.route('/data_add', methods = ['POST'])
def data_add():
    name = request.json['name']
    d_type = request.json['dataset_type']

    task_update = update_task.apply_async(args=[name, d_type])
    return jsonify({'location': url_for('api.updatestatus',
        task_id=task_update.id)}), 202

@celery.task(bind=True)
def update_task(self, name, d_type):
    new_data_dir = os.path.join(app.root_path, 'new data', name
        )
    train_dir = os.path.join(app.static_folder, 'data', 'train')
    test_dir = os.path.join(app.static_folder, 'data', 'test')
    dev_dir = os.path.join(app.static_folder, 'data', 'dev')

    if d_type == "training_dataset":
        shutil.copy(new_data_dir, train_dir)
    elif d_type == "testing_dataset":
        shutil.copy(new_data_dir, test_dir)
    elif d_type == "validation_dataset":
        shutil.copy(new_data_dir, dev_dir)
    return{}

@app.route('/data_remove', methods = ['POST'])
def data_remove():
    name = request.json['name']
    d_type = request.json['dataset_type']
    task_remove = remove_task.apply_async(args=[name, d_type
        ])
    return jsonify({'location': url_for('api.removestatus',
        task_id=task_remove.id)}), 202

@celery.task(bind=True)
def remove_task(self, name, d_type):
    if d_type == "training_dataset":
        os.remove(os.path.join(app.static_folder, 'data', 'train
            ', name))
    elif d_type == "testing_dataset":
        os.remove(os.path.join(app.static_folder, 'data', 'test
            ', name))
    elif d_type == "validation_dataset":
        os.remove(os.path.join(app.static_folder, 'data', 'dev',
            name))
    return{}

@app.route('/preprocess', methods = ['POST'])
def preprocess():
```

```
    train_name = request.form['name']
    train_data_name = request.form['t_name']
    dev_data_name = request.form['d_name']
    src = request.form['source']
    tgt = request.form['target']
    max_length = request.form['max_length']
    bpe_operations = request.form['bpe_operations']
    print bpe_operations

    task_preprocess = preprocess_task.apply_async(args=[
        train_name, train_data_name, dev_data_name, src, tgt,
        max_length, bpe_operations, current_app.config.get('
        NEMATUS_PATH')])
    return jsonify({'location': url_for('api.preprocessstatus',
        task_id=task_preprocess.id)}), 202

@celery.task(bind=True)
def preprocess_task(self, train_name, train_data_name,
    dev_data_name, src, tgt, max_length, bpe_operations,
    nematus):
    self.update_state(state='PREPROCESSING')
    log = subprocess.Popen(['bash', 'preprocess.sh', train_name
        , train_data_name, dev_data_name, src, tgt,
        max_length, bpe_operations, nematus], stdout=
        subprocess.PIPE)
    log.communicate()
    return {}

@app.route('/startTraining', methods = ['POST'])
def startTraining():
    task_train = train_task.apply_async()
    global trainId
    trainId = task_train.task_id
    return jsonify({}), 200

@app.route('/stopTraining', methods = ['POST'])
def stopTraining():
    global trainId
    task_train = AbortableAsyncResult(trainId)
    print(task_train)
    task_train.abort()
    return jsonify({}), 200

@celery.task(bind=True, base=AbortableTask)
def train_task(self):
    self.update_state(state='TRAINING')

    filename = 'logs/train.log'
    with io.open(filename, 'ab') as writer, io.open(filename, '
        rb', 1) as reader:
        process = subprocess.Popen(['bash', 'train.sh'], stdout
            =writer)
        while (process.poll() is None) and (not self.is_aborted
            ()):
            sys.stdout.write(reader.read())
            time.sleep(0.5)
        #read the remaining
        sys.stdout.write(reader.read())

        #closes process
        #process.terminate()

        #closes process tree but also prematurely kills the
            celery worker
        #if the proper way of closing in windows is
            implemented, change this code
        if self.is_aborted():
            kill_proc_tree(os.getpid())
    writer.close() #closes the writer
    reader.close() #closes the reader
    return{}

@app.route('/deployModel', methods=['POST'])
def deployModel():
    model_name = request.json['model']
    base_model = get_name(request.json['base_model'])

    task_deploy = deploy_task.apply_async(args=[model_name,
        base_model])
    return jsonify({'location': url_for('api.deploystatus',
        task_id=task_deploy.id)}), 202

@celery.task(bind=True)
def deploy_task(self, model_name, base_model):
    self.update_state(state='DEPLOYING')
    #generic name of model is : SRCTGT.npz
    if model_name != "" and base_model != "":
        generic_name = get_extension(base_model)
        base_model_config = 'model_training/' + base_model +
            ".json"
        deployed_config = 'model_deployed/' + generic_name +
            '.npz.json'
        model_location = 'model_training/' + model_name
        deployed_model = 'model_deployed/' + generic_name +
            '.npz'
        bpe_file = 'model_training/' + generic_name + ".bpe"
        bpe_file_model = 'model_deployed/' + generic_name + ".
            npz.bpe"

        shutil.copy(base_model_config, deployed_config)
        shutil.copy(model_location, deployed_model)
        shutil.copy(bpe_file, bpe_file_model)
    return{}

@app.route('/testmodel', methods=['POST'])
def testmodel():
    model_name = request.json['model']
    base_model = get_name(request.json['base_model'])
    test_name = request.json['t_name']
    src = request.json['source']
    tgt = request.json['target']
    task_test = test_task.apply_async(args=[base_model,
        model_name, test_name, src, tgt, current_app.config.get
        ('NEMATUS_PATH')])
    return jsonify({'location': url_for('api.teststatus',
        task_id=task_test.id)}), 202

@celery.task(bind=True)
def test_task(self, base_model, model_name, test_name, src, tgt,
    nematus):
    self.update_state(state='TESTING', meta={'bleuScore':
        ''})
    log = subprocess.Popen(['bash', 'test.sh', base_model,
        model_name, test_name, src, tgt, nematus], stdout=
        subprocess.PIPE, stderr=subprocess.PIPE)
    output, error = log.communicate()
    print output
    print error
    bleuScore = output.strip()
    if output.strip() == "":
        bleuScore = "0.00"
    return{'bleuScore': bleuScore}

@app.route('/testtranslate', methods=['POST'])
def testtranslate():
    input = request.json['input']
    model_type = request.json['type']
    base_model = 'model_training/' + get_name(request.json['
        base_model'])
    model = 'model_training/' + request.json['model']
    model_config = model + ".json"
    base_model_config = base_model + ".json"
    bpe_file = 'model_training/' + get_extension(base_model) +
        ".bpe"
    bpe_file_model = model + ".bpe"
    shutil.copy(base_model_config, model_config)
    shutil.copy(bpe_file, bpe_file_model)
    print input
    print model_type
    print model
    if((model_type == 'engfil') or (model_type == 'fileng')):
        task_translation = translation_task.apply_async(args=[
            input, model_type, model, current_app.config.get('
            NEMATUS_PATH')])
        return jsonify({'location': url_for('api.
            translationstatus', task_id=task_translation.id)})
            , 202

@celery.task(bind=True)
def translation_task(self, input, model_type, model, nematus):
    self.update_state(state='TRANSLATING', meta={'
        translation': ''})
    task_number = self.request.id
    log = subprocess.Popen(['bash', 'oneTranslateGlobal.sh',
        input, model_type, task_number, model, nematus],
        stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    output, error = log.communicate()
    return {'translation': output.strip()}

@app.route('/generateConfig', methods = ['POST'])
def generateConfig():
    name = request.form['name']
    optimizer = request.form['optimizer']
    dim_word = request.form['dim_word']
    dim = request.form['dim']
    n_words_src = request.form['n_words_src']
    n_words = request.form['n_words']
    lrate = request.form['lrate']
    batch_size = int(request.form['batch_size'])
    valid_batch_size = request.form['valid_batch_size']
    dispFreqCode = request.form['dispFreq'] #code
    validFreqCode = request.form['validFreq'] #code
    saveFreqCode = request.form['saveFreq'] #code
    sampleFreqCode = request.form['sampleFreq'] #code
```

```
maxlen = request.form['maxlen']
finish_after = request.form[' finish_after ']
max_epochs = request.form['max_epochs']
decay_c = request.form['decay_c']
map_decay_c = request.form['map_decay_c']
alpha_c = request.form['alpha_c']
clip_c = request.form[' clip_c ']
use_dropout = request.form['use_dropout'] #checked
dropout_embedding = request.form['dropout_embedding']
dropout_hidden = request.form['dropout_hidden']
dropout_source = request.form['dropout_source']
dropout_target = request.form['dropout_target']
src = request.form['source']
tgt = request.form['target']


dataset_size = file_len("data_training/train_data.bpe." +
    src)
epoch = dataset_size / batch_size
dispFreq = compute_epoch(epoch, dispFreqCode)
validFreq = compute_epoch(epoch, validFreqCode)
saveFreq = compute_epoch(epoch, saveFreqCode)
sampleFreq = compute_epoch(epoch, sampleFreqCode)

#Ceiling of < 1, since it is computed as integer, not float
if dispFreq == 0:
    dispFreq = 1
    dispFreqGraph = 1
else:
    dispFreqGraph = dispFreq / epoch

if validFreq == 0:
    validFreq = 1
    validFreqGraph = 1
else:
    validFreqGraph = validFreq / epoch

if saveFreq == 0:
    saveFreq = 1

if sampleFreq == 0:
    sampleFreq = 1

#initial logs
f = open("logs/train.log","w")
f.write("Name " + name + " dispFreq " + str(
    dispFreqGraph) + " validFreq " + str(validFreqGraph)
    + "\n")
f.close()

f = open("training_config.py","w") #opens file with name of
    "test.txt"
f.write("import numpy\n")
f.write("import os\n")
f.write("import sys\n\n")
f.write("SRC = '" + src + "'\n")
f.write("TGT = '" + tgt + "'\n")
f.write("DATA_DIR = 'data_training'\n")
f.write("DICT_DIR = 'dictionaries'\n\n")
f.write("#Script based on https://github.com/rsennrich/
    nematus \n")
f.write("from nematus.nmt import train\n")
f.write("if __name__ == '__main__':\n")
f.write("    validerr = train(saveto='model_training/" +
    name + "." + src + tgt + "', #Location of the Model
    \n")
f.write("                     reload_=False,\n")
f.write("                     optimizer='" + optimizer +"',
    #Optimizer\n")
f.write("                     dim_word=" + dim_word +",
    #Embedding Layer Size\n")
f.write("                     dim=" + dim +",#Hidden
    Layer Size\n")
f.write("                     lrate=" + lrate +", #Learning
     Rate\n")
f.write("                     n_words_src=" + n_words_src
    +", #Source Vocabulary Size\n")
f.write("                     n_words=" + n_words +", #
    Target Vocabulary Size\n")
f.write("                     batch_size=" + str(batch_size)
    +", #Minibatch Size\n")
f.write("                     valid_batch_size=" +
    valid_batch_size +", #Validation Batch Size\n")
f.write("                     dispFreq=" + str(dispFreq) +",
     #Loss Display Frequency\n")
f.write("                     validFreq=" + str(validFreq)
    +", #Validation Frequency\n")
f.write("                     saveFreq=" + str(saveFreq)
    +", #Save Frequency\n")
f.write("                     sampleFreq=" + str(
    sampleFreq) +", #Sample Frequency\n")
f.write("                     finish_after =" + finish_after
    + ", #Max Minibatches\n") #edit
f.write("                     max_epochs=" + max_epochs
    + ", #Max Epochs\n") #edit
f.write("                     decay_c=" + decay_c +", #L2
    Regularization Penalty\n")
f.write("                     map_decay_c=" + decay_c +",
    #L2 Regularization Penalty (Original Weights)\n")
f.write("                     alpha_c=" + alpha_c+", #
    Alignment Regularization\n")
f.write("                     clip_c =" + clip_c +", #
    Gradient Clipping Threshold\n")
f.write("                     use_dropout=" + use_dropout
    +", #Dropout\n")
f.write("                     dropout_embedding=" +
    dropout_embedding +", #For Embedding Layer\n")
f.write("                     dropout_hidden=" +
    dropout_hidden +", #For Hidden Layer\n")
f.write("                     dropout_source=" +
    dropout_source +", #For Encoder\n")
f.write("                     dropout_target=" +
    dropout_target +", #For Decoder \n")
f.write("                     maxlen=" + maxlen +", #Max
     Sentence Length\n")
f.write("                     datasets=[DATA_DIR + '/
    train_data.bpe.' + SRC, DATA_DIR + '/train_data.
    bpe.' + TGT],\n")
f.write("                     valid_datasets=[DATA_DIR +
    '/dev_data.bpe.' + SRC, DATA_DIR + '/dev_data.bpe
    .' + TGT],\n")
f.write("                     dictionaries=[DICT_DIR + '/"
     + name + ".bpe.' + SRC + '.json',DICT_DIR + '/"
    + name + ".bpe.' + TGT + '.json'],\n")
f.write("                     overwrite=False,\n")
f.write("                     external_validation_script ='./
    validate.sh')\n")
f.write("    print(validerr)\n")
f.close()

f = open("validate.sh", "w")
f.write("#/bin/sh\n\n")
f.write("# path to nematus\n")
f.write("nematus=" + current_app.config.get('
    NEMATUS_PATH') + "\n\n")
f.write("# path to moses decoder scripts\n")
f.write("moses_scripts=moses-scripts\n\n")
f.write("#theano device mode\n")
f.write("device=gpu\n\n")
f.write("#model prefix\n")
f.write("prefix=model_training/" + name + "." + src + tgt
    + "\n\n")
f.write("#source language\n")
f.write("SRC=" + src + "\n\n")
f.write("#target language\n")
f.write("TGT=" + tgt + "\n\n")
f.write("dev=data_training/dev_data.bpe.$SRC\n")
f.write("ref=data_training/dev_data.bpe.$TGT\n\n")
f.write("#theano config\n")
f.write("THEANO_FLAGS=mode=FAST_RUN,floatX=
    float32,device=$device,on_unused_input=warn python
    $nematus/nematus/translate.py \\\n")
f.write("    -m $prefix.dev.npz \\\n")
f.write("    -i $dev \\\n")
f.write("    -o $dev.output.dev \\\n")
f.write("    -k 12 -n -p 1 \n\n")
f.write("./postprocess-dev.sh < $dev.output.dev > $dev.
    output.postprocessed.dev\n\n")
f.write("## get BLEU\n")
f.write("BEST=`cat ${prefix}_best_bleu || echo 0`\n")
f.write("$moses_scripts/generic/multi-bleu.perl $ref < $dev
    .output.postprocessed.dev >> logs/" + name + "
    _bleu_scores\n")
f.write("BLEU=`$moses_scripts/generic/multi-bleu.perl
    $ref < $dev.output.postprocessed.dev | cut -f 3 -d ' '
     | cut -f 1 -d ','`\n")
f.write("BETTER=`echo \"$BLEU > $BEST\" | bc`\n\n")
f.write("if [ \"$BLEU\" = \"\" ]; then\n")
f.write("    echo \"No BLEU Score returned. Writing 0.00
    Manually...\"\n")
f.write("    echo \"BLEU = 0.00\"\n")
f.write("else\n")
f.write("    echo \"BLEU = $BLEU\"\n")
f.write("fi\n\n")
f.write("echo \"\" \n")
f.write("# save model with highest BLEU\n")
f.write("if [ \"$BETTER\" = \"1\" ]; then\n")
f.write("    echo \"new best; saving\"\n")
f.write("    echo $BLEU > ${prefix}_best_bleu\n")
f.write("    cp ${prefix}.dev.npz ${prefix}.best_bleu.npz\n")
f.write("fi\n")
f.close()

#cleans trailing /r of the shell script
```

```
    subprocess.Popen(['sed', '-i', 's/\\r$//', 'validate.sh'])

    return jsonify({}), 200

@app.route('/favicon.ico')
def favicon():
    return send_from_directory(os.path.join(app.root_path, '
        static'),
                               'favicon.ico', mimetype='image
                                   /vnd.microsoft.icon')

def file_len(fname):
    with open(fname) as f:
        for i, l in enumerate(f):
            pass
    return i + 1

def compute_epoch(epoch, code):
    if code == "e1_2":
        return epoch / 2
    elif code == "e1_4":
        return epoch / 4
    elif code == "e1_8":
        return epoch / 8
    elif code == "e":
        return epoch
    elif code == "e5_2":
        return epoch * 5 / 2
    elif code == "e5":
        return epoch * 5
    elif code == "e10":
        return epoch * 10

def get_extension(filename):
    return os.path.splitext(filename)[1].split(".")[1]

def get_name(filename):
    return os.path.splitext(filename)[0]

def kill_proc_tree(pid, including_parent=True):
    parent = psutil.Process(pid)
    children = parent.children(recursive=True)
    for child in children:
        child.kill()
    psutil.wait_procs(children, timeout=5)
    if including_parent:
        parent.kill()
        parent.wait(5)

def file_len(fname):
    i = -1
    with open(fname) as f:
        for i, l in enumerate(f):
            pass
    return i + 1

if __name__ == '__main__':
    server = WSGIServer(('127.0.0.1', 5000),
        DebuggedApplication(app))
    server.serve_forever()
```

File: neuro-mt\training_config.py

```
import numpy
import os
import sys

SRC = 'fil'
TGT = 'eng'
DATA_DIR = 'data_training'
DICT_DIR = 'dictionaries'

#Script based on https://github.com/rsennrich/nematus
from nematus.nmt import train
if __name__ == '__main__':
    validerr = train(saveto='model_training/Test1.fileng', #
        Location of the Model
                     reload_=False,
                     optimizer='adam', #Optimizer
                     dim_word=25, #Embedding Layer Size
                     dim=50,#Hidden Layer Size
                     lrate=0.01, #Learning Rate
                     n_words_src=20, #Source Vocabulary Size
                     n_words=20, #Target Vocabulary Size
                     batch_size=2, #Minibatch Size
                     valid_batch_size=2, #Validation Batch Size
                     dispFreq=2, #Loss Display Frequency
                     validFreq=10, #Validation Frequency
                     saveFreq=20, #Save Frequency
                     sampleFreq=20, #Sample Frequency
                     finish_after=10000000, #Max Minibatches
                     max_epochs=10, #Max Epochs
                     decay_c=0, #L2 Regularization Penalty
                     map_decay_c=0, #L2 Regularization Penalty
                         (Original Weights)
                     alpha_c=0, #Alignment Regularization
                     clip_c=1, #Gradient Clipping Threshold
                     use_dropout=False, #Dropout
                     dropout_embedding=0.2, #For Embedding
                         Layer
                     dropout_hidden=0.2, #For Hidden Layer
                     dropout_source=0, #For Encoder
                     dropout_target=0, #For Decoder
                     maxlen=10, #Max Sentence Length
                     datasets=[DATA_DIR + '/train_data.bpe.' +
                         SRC, DATA_DIR + '/train_data.bpe.'
                         + TGT],
                     valid_datasets=[DATA_DIR + '/dev_data.
                         bpe.' + SRC, DATA_DIR + '/
                         dev_data.bpe.' + TGT],
                     dictionaries=[DICT_DIR + '/Test1.bpe.' +
                         SRC + '.json',DICT_DIR + '/Test1.
                         bpe.' + TGT + '.json'],
                     overwrite=False,
                     external_validation_script='./validate.sh')
    print(validerr)
```

File: neuro-mt\validate.sh

```
#/bin/sh

# path to nematus
nematus=C:/Users/Alfred/Anaconda2/Lib/site-packages/
    nematus

# path to moses decoder scripts
moses_scripts=moses-scripts

#theano device mode
device=gpu

#model prefix
prefix=model_training/Test1.fileng

#source language
SRC=fil

#target language
TGT=eng

dev=data_training/dev_data.bpe.$SRC
ref=data_training/dev_data.bpe.$TGT

#theano config
THEANO_FLAGS=mode=FAST_RUN,floatX=float32,device=
    $device,on_unused_input=warn python $nematus/nematus/
    translate.py \
    -m $prefix.dev.npz \
    -i $dev \
    -o $dev.output.dev \
    -k 12 -n -p 1

./postprocess-dev.sh < $dev.output.dev > $dev.output.
    postprocessed.dev

## get BLEU
BEST=`cat ${prefix}_best_bleu || echo 0`
$moses_scripts/generic/multi-bleu.perl $ref < $dev.output.
    postprocessed.dev >> logs/Test1_bleu_scores
BLEU=`$moses_scripts/generic/multi-bleu.perl $ref < $dev.
    output.postprocessed.dev | cut -f 3 -d ' ' | cut -f 1 -d
    ','`
BETTER=`echo "$BLEU > $BEST" | bc`

if [ "$BLEU" = "" ]; then
  echo "No BLEU Score returned. Writing 0.00 Manually..."
  echo "BLEU = 0.00"
else
  echo "BLEU = $BLEU"
fi

echo ""
# save model with highest BLEU
if [ "$BETTER" = "1" ]; then
  echo "new best; saving"
  echo $BLEU > ${prefix}_best_bleu
  cp ${prefix}.dev.npz ${prefix}.best_bleu.npz
fi
```

File: neuro-mt\static\scripts_bootstrap.js

```
function request_translation() {
    var source = document.getElementById('language-one-
        value').value;
    var target = document.getElementById('language-two-
        value').value;
    var input = document.getElementById('input').value;
    // send ajax POST request to start background job
    $.ajax({
        type: 'POST',
        url: '/translate',
        data: JSON.stringify({ source: source, target: target,
            input: input}),
        contentType: "application/json; charset=utf-8",
        success: function(data, status, request) {
            document.getElementById('translate_button').
                disabled = true;
            document.getElementById('suggest_button').disabled
                = true;
            document.getElementById('clear_button').disabled =
                true;
            document.getElementById('switcher').disabled =
                true;
            status_url = data['location'];
            update_translation(status_url);
        },
        error: function() {
            alert('Unexpected error');
        }
    });
}

function update_translation(status_url) {
    // send GET request to status URL

    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            document.getElementById('translation').value = '
                System Error: ' + data['translation'];
        }
        else if (data['state'] == 'SUCCESS'){
            document.getElementById('translation').value =
                data['translation'];
            document.getElementById('translate_button').
                disabled = false;
            document.getElementById('suggest_button').disabled
                = false;
            document.getElementById('clear_button').disabled =
                false;
            document.getElementById('switcher').disabled =
                false;
        }
        else {
            // rerun in 3 seconds
            document.getElementById('translation').value = "
                Translating...";
            setTimeout(function() {
                update_translation(status_url);
            }, 3000);
        }
    });
}

function switch_language(){
    if(document.getElementById('translation').value != ""){
        alert("Clear everything first");
    }
    else{
        var val1 = document.getElementById('language-one').
            innerHTML;
        var val1_value = document.getElementById('language-
            one-value').value;
        var val2 = document.getElementById('language-two').
            innerHTML;
        var val2_value = document.getElementById('language-
            two-value').value;

        document.getElementById('language-one').innerHTML
            = val2;
        document.getElementById('language-one-value').value
            = val2_value;
        document.getElementById('language-two').innerHTML
            = val1;
        document.getElementById('language-two-value').value
            = val1_value;
    }
}

function clear_textarea(){
    document.getElementById('input').value = "";
    document.getElementById('translation').value = "";
}

function enable_suggest(){
    document.getElementById('translation').readOnly = false;
    document.getElementById('suggest_button').style.display =
        "none";
    document.getElementById('suggest_cancel').style.display = "
        block";
    document.getElementById('suggest_submit').style.display =
        "block";
    document.getElementById('translation').focus();
}

function disable_suggest(){
    document.getElementById('translation').readOnly = true;
    document.getElementById('suggest_button').style.display =
        "block";
    document.getElementById('suggest_cancel').style.display = "
        none";
    document.getElementById('suggest_submit').style.display =
        "none";
}

function send_suggestion(){
    var english = "";
    var filipino = "";
    var lang_one = document.getElementById('language-one-
        value').value;
    var lang_two = document.getElementById('language-two-
        value').value;
    if(lang_one == "fil" && lang_two == "eng"){
        filipino = document.getElementById('input').value;
        english = document.getElementById('translation').value
            ;
    }
    else if(lang_one == "eng" && lang_two == "fil"){
        english = document.getElementById('input').value;
        filipino = document.getElementById('translation').
            value;
    }
    $.ajax({
        type: 'POST',
        url: '/suggest',
        data: JSON.stringify({ eng: english, fil: filipino }),
        contentType: "application/json; charset=utf-8",
        success: function(data, status, request) {
            alert('Thank you for the suggestion!');
        },
        error: function() {
            alert('Failed to send suggestion, thanks for the
                effort.');
        }
    });
    disable_suggest();
    clear_textarea();
}
```

File: neuro-mt\static\scripts_data.js

```
$(document).ready(function() {
    $('select').material_select();

    $('.tooltipped').tooltip({delay: 50});

    $('ul.tabs').tabs();

    $("button[id = 'save']").click(function(){
        var address = $(this).attr("href");
        if(document.getElementById("suggestions_count").
            innerHTML == "0"){
            $toastContent = $('<span> No suggestions to save
                .</span>');
            Materialize.toast($toastContent, 5000);
        }
        else{
            $.get(address,
            function(data,status){
                if(status == 'success'){
                    $toastContent = $('<span> Saving...</span
                        >');
                    Materialize.toast($toastContent, 5000);
                    update_save(data['location']);
                }
                else{
                    $toastContent = $('<span> Saved failed</
                        span>');
                    Materialize.toast($toastContent, 5000);
                }
            });
        }
    });

    $("button[id = 'addTrain']").click(function(){
```

```
    var address = $(this).attr("href");
    var name = document.getElementById("tnd_append").
        value;
    if(name == ""){
        $toastContent = $('<span> No Training Data
            Selected</span>');
        Materialize.toast($toastContent, 5000);
    }
    else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({ name: name,
                dataset_type: 'training_dataset'}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                $toastContent = $('<span> Updating
                    Training Data</span>');
                Materialize.toast($toastContent, 5000);
                update_update(data['location'], 'Training
                    Data');
            },
            error: function() {
                $toastContent = $('<span> Failed to
                    Update Training Data</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    }
});

$("button[id = 'addTest']").click(function(){
    var address = $(this).attr("href");
    var name = document.getElementById("ttd_append").
        value;
    if(name == ""){
        $toastContent = $('<span> No Testing Data
            Selected</span>');
        Materialize.toast($toastContent, 5000);
    }
    else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({ name: name,
                dataset_type: 'testing_dataset'}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                $toastContent = $('<span> Updating
                    Testing Data</span>');
                Materialize.toast($toastContent, 5000);
                update_update(data['location'], 'Testing
                    Data');
            },
            error: function() {
                $toastContent = $('<span> Failed to
                    Update Testing Data</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    }
});

$("button[id = 'addDev']").click(function(){
    var address = $(this).attr("href");
    var name = document.getElementById("vdd_append").
        value;
    if(name == ""){
        $toastContent = $('<span> No Validation Data
            Selected</span>');
        Materialize.toast($toastContent, 5000);
    }
    else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({ name: name,
                dataset_type: 'validation_dataset'}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                $toastContent = $('<span> Updating
                    Validation Data</span>');
                Materialize.toast($toastContent, 5000);
                update_update(data['location'], 'Validation
                    Data');
            },
            error: function() {
                $toastContent = $('<span> Failed to
                    Update Validation Data</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    }
});

$("button[id = 'removeTrain']").click(function(){
    var address = $(this).attr("href");
    var name = document.getElementById("tnd_remove").
        value;
    if(name == ""){
        $toastContent = $('<span> No Training Data
            Selected</span>');
        Materialize.toast($toastContent, 5000);
    }
    else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({ name: name,
                dataset_type: 'training_dataset'}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                $toastContent = $('<span> Removing
                    Training Data</span>');
                Materialize.toast($toastContent, 5000);
                update_delete(data['location'], 'Training
                    Data');
            },
            error: function() {
                $toastContent = $('<span> Failed to
                    Remove Training Data</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    }
});

$("button[id = 'removeTest']").click(function(){
    var address = $(this).attr("href");
    var name = document.getElementById("ttd_remove").
        value;
    if(name == ""){
        $toastContent = $('<span> No Testing Data
            Selected</span>');
        Materialize.toast($toastContent, 5000);
    }
    else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({ name: name,
                dataset_type: 'testing_dataset'}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                $toastContent = $('<span> Removing
                    Testing Data</span>');
                Materialize.toast($toastContent, 5000);
                update_delete(data['location'], 'Testing
                    Data');
            },
            error: function() {
                $toastContent = $('<span> Failed to
                    Remove Testing Data</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    }
});

$("button[id = 'removeDev']").click(function(){
    var address = $(this).attr("href");
    var name = document.getElementById("vdd_remove").
        value;
    if(name == ""){
        $toastContent = $('<span> No Validation Data
            Selected</span>');
        Materialize.toast($toastContent, 5000);
    }
        else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({ name: name,
                dataset_type: 'validation_dataset'}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                $toastContent = $('<span> Removing
                    Validation Data</span>');
                Materialize.toast($toastContent, 5000);
                update_delete(data['location'], 'Validation
                    Data');
            },
            error: function() {
                $toastContent = $('<span> Failed to
```

```
                        Remove Validation Data</span>');
                    Materialize.toast($toastContent, 5000);
                }
            });
        }
    });
});

function update_save(status_url){
    // send GET request to status URL

    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            $toastContent = $('<span> Save failed.</span>');
            Materialize.toast($toastContent, 5000);
        }
        else if (data['state'] == 'SUCCESS'){
            $toastContent = $('<span> Saved to suggestions
                folder.</span>');
            document.getElementById("suggestions_count").
                innerHTML = "0";
            Materialize.toast($toastContent, 5000);
        }
        else {
            // rerun in 1 second
            setTimeout(function() {
                update_save(status_url);
            }, 1000);
        }
    });
}

//type = Training Data/ Testing Data/ Validation Data
function update_update(status_url, type){
    // send GET request to status URL

    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            $toastContent = $('<span> Failed To Update ' +
                type + '.</span>');
            Materialize.toast($toastContent, 5000);
        }
        else if (data['state'] == 'SUCCESS'){
            $toastContent = $('<span> ' + type + ' Updated
                .</span>');
            Materialize.toast($toastContent, 5000);
        }
        else {
            // rerun in 1 second
            setTimeout(function() {
                update_update(status_url,type);
            }, 1000);
        }
    });
}

//type = Training Data/ Testing Data/ Validation Data
function update_delete(status_url, type){
    // send GET request to status URL

    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            $toastContent = $('<span> Failed To Remove ' +
                type + '.</span>');
            Materialize.toast($toastContent, 5000);
        }
        else if (data['state'] == 'SUCCESS'){
            $toastContent = $('<span> ' + type + ' Removed
                .</span>');
            Materialize.toast($toastContent, 5000);
        }
        else {
            // rerun in 1 second
            setTimeout(function() {
                update_delete(status_url,type);
            }, 1000);
        }
    });
}
```

File: neuro-mt\static\scripts_index.js

```
$(document).ready(function() {
    $("button[id = 'stopTraining']").click(function(){
        var address = $(this).attr("href");
        this.disabled = true;
        $.post(address,
        function(data,status){
            if(status == 'success'){
                $toastContent = $('<span> Training
                    successfully stopped.</span>');
                Materialize.toast($toastContent, 5000);
            }
            else{
                $toastContent = $('<span> Failed to stop
                    training.</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    });
});
```

File: neuro-mt\static\scripts_materialize.js

```
$(document).ready(function() {
    $('select').material_select();

    $('.tooltipped').tooltip({delay: 50});

    $('ul.tabs').tabs();

    $('.carousel.carousel-slider').carousel({fullWidth: true});

    setInterval(function(){
            $('.carousel.carousel-slider').carousel('next');
    }, 5000);

    $('.parallax').parallax();

    $('.scrollspy').scrollSpy();

    $('.toc-wrapper').pushpin({
        top: 0,
        bottom: 2200,
        offset: 100
    });
});
```

File: neuro-mt\static\scripts_test.js

```
$(document).ready(function() {
    $('select').material_select();
    $('.tooltipped').tooltip({delay: 50});
    $("button[id = 'test']").click(function(){
        var address = $(this).attr("href");

        var t_name = get_name(document.getElementById("
            td_src").value);
        var source = get_extension(document.getElementById("
            td_src").value);
        var target = get_extension(document.getElementById("
            td_tgt").value);
        var model = document.getElementById("test_model").
            value;
        var base_model = document.getElementById("
            base_model").value;

        if((get_extension(get_name(base_model)) != "engfil" &&
            get_extension(get_name(base_model)) != "fileng" )
            || base_model == ""){
            $toastContent = $('<span> Invalid Base Model</
                span>');
            Materialize.toast($toastContent, 5000);
        }
        else if(model == "" || ((get_name(get_name(model)) !=
            get_name(get_name(base_model))) &&
            get_extension(get_name(model)) != "best_bleu") ||
        ((get_name(get_name(get_name(model))) != get_name(
            get_name(base_model))) && get_extension(
            get_name(model)) == "best_bleu")){
            $toastContent = $('<span> Invalid Model to be
                Tested</span>');
            Materialize.toast($toastContent, 5000);
        }
        else if(document.getElementById("td_src").value ==
            ""){
            $toastContent = $('<span> No Testing Source
                Dataset Selected</span>');
            Materialize.toast($toastContent, 5000);
        }
        else if(document.getElementById("td_tgt").value ==
            ""){
            $toastContent = $('<span> No Testing Target
                Dataset Selected</span>');
            Materialize.toast($toastContent, 5000);
        }
        else{
            $.ajax({
                type: 'POST',
                url: address,
                data: JSON.stringify({ t_name: t_name, source:
```

```
                    source, target: target, model: model,
                    base_model: base_model}),
                contentType: "application/json; charset=utf-8",
                success: function(data, status, request) {
                    document.getElementById('test').disabled =
                        true;
                    status_url = data['location'];
                    $toastContent = $('<span> Testing</span
                        >');
                    Materialize.toast($toastContent, 5000);
                    update_score(status_url);
                },
                error: function() {
                    document.getElementById('test').disabled =
                        false;
                    $toastContent = $('<span> An error
                        occured</span>');
                    Materialize.toast($toastContent, 5000);
                }
            });
    }
});

$("button[id = 'deploy']").click(function(){
    var address = $(this).attr("href");
    var model = document.getElementById("test_model").
        value;
    var base_model = document.getElementById("
        base_model").value;

    if((get_extension(get_name(base_model)) != "engfil" &&
        get_extension(get_name(base_model)) != "fileng" )
        || base_model == ""){
        $toastContent = $('<span> Invalid Base Model</
            span>');
        Materialize.toast($toastContent, 5000);
    }
    else if(model == "" || ((get_name(get_name(model)) !=
        get_name(get_name(base_model))) &&
        get_extension(get_name(model)) != "best_bleu") ||
    ((get_name(get_name(get_name(model))) != get_name(
        get_name(base_model))) && get_extension(
        get_name(model)) == "best_bleu")){
        $toastContent = $('<span> Invalid Model to be
            Tested</span>');
        Materialize.toast($toastContent, 5000);
    }
    else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({model: model, base_model
                : base_model}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                $toastContent = $('<span> Deploying
                    model...</span>');
                Materialize.toast($toastContent, 5000);
                update_deploy(data['location'])
            },
            error: function() {
                document.getElementById('test').disabled =
                    false;
                $toastContent = $('<span> Failed to deploy
                    model.</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    }
});

$("button[id = 'testTranslate']").click(function(){
    var address = $(this).attr("href");

    var input = document.getElementById("input").value;
    var type = get_extension(get_name((document.
        getElementById("base_model").value)));
    var model = document.getElementById("test_model").
        value;
    var base_model = document.getElementById("
        base_model").value;

    if((get_extension(get_name(base_model)) != "engfil" &&
        get_extension(get_name(base_model)) != "fileng" )
        || base_model == ""){
        $toastContent = $('<span> Invalid Base Model</
            span>');
        Materialize.toast($toastContent, 5000);
    }
    else if(model == "" || ((get_name(get_name(model)) !=
        get_name(get_name(base_model))) &&
        get_extension(get_name(model)) != "best_bleu") ||
        ((get_name(get_name(get_name(model))) != get_name(
        get_name(base_model))) && get_extension(
        get_name(model)) == "best_bleu")){
        $toastContent = $('<span> Invalid Model to be
            Tested</span>');
        Materialize.toast($toastContent, 5000);
    }
    else{
        $.ajax({
            type: 'POST',
            url: address,
            data: JSON.stringify({ input: input, type: type,
                model: model, base_model: base_model}),
            contentType: "application/json; charset=utf-8",
            success: function(data, status, request) {
                document.getElementById('testTranslate').
                    disabled = true;
                status_url = data['location'];
                update_translation(status_url);
            },
            error: function() {
                document.getElementById('testTranslate').
                    disabled = false;
                $toastContent = $('<span> An error
                    occured</span>');
                Materialize.toast($toastContent, 5000);
            }
        });
    }
});

function update_score(status_url){
    // send GET request to status URL

    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            document.getElementById('bleu_score').innerHTML
                = 'System Error: ' + data['bleuScore'];
        }
        else if (data['state'] == 'SUCCESS'){
            document.getElementById('bleu_score').innerHTML
                = data['bleuScore'];
            document.getElementById('test').disabled = false;
        }
        else {
            // rerun in 1 second
            document.getElementById('bleu_score').innerHTML
                = "Testing...";
            setTimeout(function() {
                update_score(status_url);
            }, 1000);
        }
    });
}

function update_deploy(status_url){
    // send GET request to status URL

    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            $toastContent = $('<span> Failed to deploy model
                .</span>');
            Materialize.toast($toastContent, 5000);
        }
        else if (data['state'] == 'SUCCESS'){
            $toastContent = $('<span> Model Deployed.</span
                >');
            Materialize.toast($toastContent, 5000);
        }
        else {
            // rerun in 1 second
            setTimeout(function() {
                update_deploy(status_url);
            }, 1000);
        }
    });
}

function update_translation(status_url){
    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            document.getElementById('translation').innerHTML
                = 'System Error: ' + data['translation'];
        }
        else if (data['state'] == 'SUCCESS'){
            document.getElementById('translation').innerHTML
                = data['translation'];
            document.getElementById('testTranslate').disabled
```

```
                = false;
            }
            else {
                // rerun in 1 second
                document.getElementById('translation').innerHTML
                    = "Translating...";
                setTimeout(function() {
                    update_translation(status_url);
                }, 1000);
            }
            $('#translation').trigger('autoresize');
            document.getElementById('translation').focus();
            document.getElementById('translation').blur();
        });
}


function notify_language(e){
    var val = e.value;
    var $toastContent;
    var src_select = document.getElementById("td_src");
    var tgt_select = document.getElementById("td_tgt");
    var model_select = document.getElementById("test_model");
    var base_select = document.getElementById("base_model");

    if(e.id == "base_model"){
        if((get_extension(get_name(base_select.value)) != "
            engfil" && get_extension(get_name(base_select.
            value)) != "fileng" )
            || base_select.value == ""){
            $toastContent = $('<span> Invalid Base Model</
                span>');
            document.getElementById("base_model").
                selectedIndex = 0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_name(get_name(base_select.value)) !=
            get_name(get_name(model_select.value)) &&
            get_extension(get_name(model_select.value)) != "
            best_bleu" ||
        get_name(get_name(base_select.value)) != get_name(
            get_name(get_name(model_select.value))) &&
            get_extension(get_name(model_select.value)) != "
            best_bleu"){
            $toastContent = $('<span> Base Model Changed</
                span>');
            document.getElementById("test_model").
                selectedIndex = 0;
            document.getElementById("td_src").selectedIndex =
                0;
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
    }
    else if(e.id == "test_model"){
        $toastContent = $('<span> Model ' + model_select.
            value + ' will be tested</span>');
        if(model_select.value == "" || ((get_name(get_name(
            model_select.value)) != get_name(get_name(
            base_select.value))) && get_extension(get_name(
            model_select.value)) != "best_bleu") ||
        ((get_name(get_name(get_name(model_select.value))) !=
            get_name(get_name(get_name(base_select.value))) &&
            get_extension(get_name(model_select.value)) == "
            best_bleu")){
            $toastContent = $('<span> Invalid Model to be
                Tested</span>');
            document.getElementById("test_model").
                selectedIndex = 0;
            $('select').material_select(); //Updates Material
                Select
        }
    }
    else if(e.id == "td_src"){
        var val2 = tgt_select.value;
        if(base_model.selectedIndex == 0){
            $toastContent = $('<span> Select a Base Model
                first</span>');
            document.getElementById("td_src").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(model_select.selectedIndex == 0){
            $toastContent = $('<span> Select a Model first</
                span>');
            document.getElementById("td_src").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if((get_extension(val) != get_extension(get_name(
            base_select.value))).substring(0,3))){
            $toastContent = $('<span> Invalid source language
                </span>');
            document.getElementById("td_src").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if((tgt_select.selectedIndex != 0) && (get_name(
            val) != get_name(val2))){
            $toastContent = $('<span> Source dataset changed
                </span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) == "fil"){
            $toastContent = $('<span> Source language set to
                Filipino</span>');
        }
        else{
            $toastContent = $('<span> Source language set to
                English</span>');
        }
    }
    else if(e.id == "td_tgt"){
        var val2 = src_select.value;
        if(base_select.selectedIndex == 0){
            $toastContent = $('<span> Select a Base Model
                first</span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(model_select.selectedIndex == 0){
            $toastContent = $('<span> Select a Model first</
                span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(src_select.selectedIndex == 0){
            $toastContent = $('<span> Select a Testing source
                language first</span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if((get_extension(val) != get_extension(get_name(
            base_select.value))).substring(3))){
            $toastContent = $('<span> Invalid target language
                </span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_name(val) != get_name(val2)){
            $toastContent = $('<span> Invalid target dataset
                </span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) == "fil"){
            $toastContent = $('<span> Target language set to
                Filipino</span>');
        }
        else{
            $toastContent = $('<span> Target language set to
                English</span>');
        }
    }
    Materialize.toast($toastContent, 5000);
}

function get_extension(name){
    return name.slice((name.lastIndexOf(".") - 1 >>> 0) + 2);
}

function get_name(name){
    return name.substr(0, name.lastIndexOf('.'));
}
```

File: neuro-mt\static\scripts_train.js

```
$(document).ready(function() {
    $('select').material_select();

    $('ul.tabs').tabs();

    $("a[href = '#advanced']").click(function() {
        toggle_regularization('True');
    });

    $("a[href = '#normal']").click(function() {
        toggle_regularization('False');
    });

    $('.tooltipped').tooltip({delay: 50});

    $('.collapsible').collapsible();

    $("button[name = 'preprocess']").click(function(){
        var address = $(this).attr("href");
        name = document.getElementById("name").value;
        td_src = document.getElementById("td_src").value;
        td_tgt = document.getElementById("td_tgt").value;
        dd_src = document.getElementById("dd_src").value;
        dd_tgt = document.getElementById("dd_tgt").value;
        maxlen = document.getElementById("maxlen").value;
        bpe_operations = document.getElementById("
            bpe_operations").value;

        if(name == "" || name.toLowerCase() == "no ongoing
            training"){
            $toastContent = $('<span> Invalid Name</span>')
                ;
        }
        else if(td_src == ""){
            $toastContent = $('<span> No Training Source
                Dataset Selected</span>');
        }
        else if(td_tgt == ""){
            $toastContent = $('<span> No Training Target
                Dataset Selected</span>');
        }
        else if(dd_src == ""){
            $toastContent = $('<span> No Validation Source
                Dataset Selected</span>');
        }
        else if(dd_tgt == ""){
            $toastContent = $('<span> No Validation Target
                Dataset Selected</span>');
        }
        else if(maxlen == "" || maxlen < 1){
            $toastContent = $('<span> Invalid Maximum
                Sentence Length</span>');
        }
        else if(bpe_operations == "" || bpe_operations < 0){
            $toastContent = $('<span> Invalid BPE Merge
                Operations</span>');
        }
        else{
            $.post(address,
            {
            name: name,
            t_name: get_name(td_src),
            d_name: get_name(dd_src),
            source: get_extension(td_src),
            target: get_extension(td_tgt),
            max_length: maxlen,
            bpe_operations: bpe_operations
            },
            function(data,status){
                if(status == 'success'){
                    $toastContent = $('<span> Preparing Data
                        ...</span>');
                    update_preprocess(data['location']);
                    Materialize.toast($toastContent, 5000);
                }
                else{
                    $toastContent = $('<span> Failed to
                        prepare data.</span>');
                    Materialize.toast($toastContent, 5000);
                }
            });
        }
        Materialize.toast($toastContent, 5000);

    });

    $("button[name = 'generateConfig']").click(function(){
        var dropout_bool;
        if(document.getElementById("use_dropout").checked){
            dropout_bool = 'True';
        }
        else{
            dropout_bool = 'False';
        }
        var address = $(this).attr("href");
        var all_valid = false;
        name = document.getElementById("name").value;
        optimizer = document.getElementById("optimizer").
            value;
        dim_word = document.getElementById("dim_word").
            value;
        dim = document.getElementById("dim").value;
        n_words_src = document.getElementById("n_words_src").
            value;
        n_words = document.getElementById("n_words").value;
        lrate = document.getElementById("lrate").value;
        batch_size = document.getElementById("batch_size").
            value;
        valid_batch_size = document.getElementById("
            valid_batch_size").value;
        dispFreq = document.getElementById("dispFreq").value;
        validFreq = document.getElementById("validFreq").
            value;
        saveFreq = document.getElementById("saveFreq").value;
        sampleFreq = document.getElementById("sampleFreq").
            value;
        maxlen = document.getElementById("maxlen").value;
        finish_after = document.getElementById("finish_after")
            .value;
        max_epochs = document.getElementById("max_epochs")
            .value;
        decay_c = document.getElementById("decay_c").value;
        map_decay_c = document.getElementById("map_decay_c
            ").value;
        alpha_c = document.getElementById("alpha_c").value;
        clip_c = document.getElementById("clip_c").value;
        use_dropout = dropout_bool;
        dropout_embedding = document.getElementById("
            dropout_embedding").value;
        dropout_hidden = document.getElementById("
            dropout_hidden").value;
        dropout_source = document.getElementById("
            dropout_source").value;
        dropout_target = document.getElementById("
            dropout_target").value;
        td_src = document.getElementById("td_src").value;
        td_tgt = document.getElementById("td_tgt").value;

        regularization_mechanisms_disabled = document.
            getElementById("decay_c").disabled;

        if(name == "" || name.toLowerCase() == "no ongoing
            training"){
            $toastContent = $('<span> No Training Name
                Input</span>');
            Materialize.toast($toastContent, 5000);
            $toastContent = $('<span> Make sure it is coherent
                 with the prepared data</span>');
        }
        else if(optimizer == ""){
            $toastContent = $('<span> No Optimizer Selected
                </span>');
        }
        else if(dim_word == "" || dim_word < 1){
            $toastContent = $('<span> Invalid Embedding
                Layer Size</span>');
        }
        else if(dim == "" || dim < 1){
            $toastContent = $('<span> Invalid Hidden Layer
                Size</span>');
        }
        else if(lrate == "" || lrate > 1 || lrate < 0.00001){
            $toastContent = $('<span> Invalid Learning Rate
                </span>');
        }
        else if(n_words_src == "" || n_words_src < 10){
            $toastContent = $('<span> Invalid Source
                Vocabulary Size</span>');
        }
        else if(n_words == "" || n_words < 10){
            $toastContent = $('<span> Invalid Target
                Vocabulary Size</span>');
        }
        else if(batch_size == "" || batch_size < 1){
            $toastContent = $('<span> Invalid Minibatch Size
                </span>');
        }
        else if(valid_batch_size == "" || valid_batch_size < 1)
            {
            $toastContent = $('<span> Invalid Validation
                Minibatch Size</span>');
        }
```

```
else if(dispFreq == ""){
    $toastContent = $('<span> No Loss Display
        Frequency Selected</span>');
}
else if(validFreq == ""){
    $toastContent = $('<span> No Validation
        Frequency Selected</span>');
}
else if(saveFreq == ""){
    $toastContent = $('<span> No Save Frequency
        Selected</span>');
}
else if(sampleFreq == ""){
    $toastContent = $('<span> No Sample Frequency
        Selected</span>');
}
else if( finish_after == "" || finish_after < 1){
    $toastContent = $('<span> Invalid Max
        Minibatches</span>');
}
else if(max_epochs == "" || max_epochs < 1){
    $toastContent = $('<span> Invalid Max Epochs</
        span>');
}
else if(td_src == ""){
    $toastContent = $('<span> No Training Source
        Dataset Selected</span>');
    Materialize.toast($toastContent, 5000);
    $toastContent = $('<span> If it is already
        preprocessed, make sure its coherent</span
        >');
}
else if(td_tgt == ""){
    $toastContent = $('<span> No Training Target
        Dataset Selected</span>');
    Materialize.toast($toastContent, 5000);
    $toastContent = $('<span> If it is already
        preprocessed, make sure its coherent</span
        >');
}
else if(regularization_mechanisms_disabled == false){
    if(decay_c == "" || (decay_c < 0.0000001 &&
        decay_c > 0) || decay_c < 0){
        $toastContent = $('<span> Invalid L2
            Regularization Penalty</span>');
    }
    else if(map_decay_c == "" || (map_decay_c <
        0.0000001 && map_decay_c > 0) ||
        map_decay_c < 0){
        $toastContent = $('<span> Invalid L2
            Regularization Penalty (Original Weights)
            </span>');
    }
    else if(alpha_c == "" || (alpha_c < 0.0000001 &&
        alpha_c > 0) || alpha_c < 0){
        $toastContent = $('<span> Invalid Alignment
            Regularization</span>');
    }
    else if( clip_c == "" || (clip_c < 0.0000001 &&
        clip_c > 0) || clip_c < 0){
        $toastContent = $('<span> Invalid Gradient
            Clipping Threshold</span>');
    }
    else if(dropout_bool == 'True'){
        if(dropout_embedding == "" || (
            dropout_embedding < 0.001 &&
            dropout_embedding > 0)){
            $toastContent = $('<span> Invalid Dropout
                for Input Embeddings</span>');
        }
        else if(dropout_hidden == "" || (
            dropout_hidden < 0.001 &&
            dropout_hidden > 0)){
            $toastContent = $('<span> Invalid Dropout
                for Hidden Layer</span>');
        }
        else if(dropout_source == "" || (dropout_source
            < 0.001 && dropout_source > 0)){
            $toastContent = $('<span> Invalid Dropout
                for Encoder (Source Words)</span
                >');
        }
        else if(dropout_target == "" || (dropout_target
            < 0.001 && dropout_target > 0)){
            $toastContent = $('<span> Invalid Dropout
                for Decoder (Target Words)</span
                >');
        }
        else{
            all_valid = true;
            $toastContent = $('<span> All Valid</
                span>');
        }
    }
    else{
        dropout_embedding = 0.2;
        dropout_hidden = 0.2;
        dropout_source = 0;
        dropout_target = 0;
        all_valid = true;
        $toastContent = $('<span> All Valid</span>')
            ;
    }
}
else{
    decay_c = 0;
    map_decay_c = 0;
    alpha_c = 0;
    clip_c = 1.0;
    use_dropout = 'False';
    dropout_embedding = 0.2;
    dropout_hidden = 0.2;
    dropout_source = 0;
    dropout_target = 0;
    all_valid = true;
    $toastContent = $('<span> All Valid</span>');
}

if( all_valid == true){
    $.post(address,
    {
    name: name,
    optimizer: optimizer,
    dim_word: dim_word,
    dim: dim,
    n_words_src: n_words_src,
    n_words: n_words,
    lrate: lrate,
    batch_size: batch_size,
    valid_batch_size: valid_batch_size,
    dispFreq: dispFreq,
    validFreq: validFreq,
    saveFreq: saveFreq,
    sampleFreq: sampleFreq,
    maxlen: maxlen,
    finish_after: finish_after,
    max_epochs: max_epochs,
    decay_c: decay_c,
    map_decay_c: map_decay_c,
    alpha_c: alpha_c,
    clip_c: clip_c,
    use_dropout: use_dropout,
    dropout_embedding: dropout_embedding,
    dropout_hidden: dropout_hidden,
    dropout_source: dropout_source,
    dropout_target: dropout_target,
    source: get_extension(td_src),
    target: get_extension(td_tgt)
    },
    function(data,status){
        //alert("Data: " + data + "\nStatus: " +
            status);
        if(status == 'success'){
            $toastContent = $('<span> Configuration
                Generated</span>');
            Materialize.toast($toastContent, 5000);
        }
        else{
            $toastContent = $('<span> Failed to
                generate configuration</span>');
            Materialize.toast($toastContent, 5000);
        }
    });
}
Materialize.toast($toastContent, 5000);
});

$("button[name = 'startTraining']").click(function(){
    var address = $(this).attr("href");
    $.post(address,{},
    function(data,status){
        if(status == 'success'){
            $toastContent = $('<span> Training has
                Started</span>');
            Materialize.toast($toastContent, 5000);
        }
        else{
            $toastContent = $('<span> Training did not
                start</span>');
            Materialize.toast($toastContent, 5000);
        }
    });
});
});
```

```
function update_preprocess(status_url){
    // send GET request to status URL

    $.getJSON(status_url, function(data) {
        // update UI
        if (data['state'] == 'FAILURE'){
            $toastContent = $('<span> Failed to prepare data
                .</span>');
            Materialize.toast($toastContent, 5000);
        }
        else if (data['state'] == 'SUCCESS'){
            $toastContent = $('<span> Data successfully
                prepared.</span>');
            Materialize.toast($toastContent, 5000);
        }
        else {
            // rerun in 1 second
            setTimeout(function() {
                update_preprocess(status_url);
            }, 1000);
        }
    });
}

function notify_file_length (name, type){
    address = document.getElementById("getFileLength").href;
    $.post(address,{
        name: name,
        type: type
    },
    function(data,status){
        //alert("Data: " + data['count'] + "\nStatus: " +
            status);
        if(status == "success"){
            $toastContent = $('<span> Length is ' + data['
                count'] + '</span>');
            Materialize.toast($toastContent, 5000);
        }
        else{
            $toastContent = $('<span> Failed to get length of
                file</span>');
            Materialize.toast($toastContent, 5000);
        }
    });
}

function toggle_dropout(e){
    if(e.checked == true){
        document.getElementById("dropout_embedding").
            disabled = false;
        document.getElementById("dropout_hidden").disabled =
            false;
        document.getElementById("dropout_source").disabled =
            false;
        document.getElementById("dropout_target").disabled =
            false;
    }
    else{
        document.getElementById("dropout_embedding").
            disabled = true;
        document.getElementById("dropout_hidden").disabled =
            true;
        document.getElementById("dropout_source").disabled =
            true;
        document.getElementById("dropout_target").disabled =
            true;
    }
}

function toggle_regularization (bool){
    if(bool == 'True'){
        //document.getElementById("regularizationSection").
            display = "block";
        document.getElementById("decay_c").disabled = false;
        document.getElementById("map_decay_c").disabled =
            false;
        document.getElementById("alpha_c").disabled = false;
        document.getElementById("clip_c").disabled = false;
        document.getElementById("use_dropout").disabled =
            false;
        document.getElementById("use_dropout").onchange();
            //adjusts dropout input = number
        document.getElementById("arch_image").src =
            document.getElementById("arch_image_advanced
            ").href;
    }
    else{
        //document.getElementById("regularizationSection").
            display = "none";
        document.getElementById("decay_c").disabled = true;
        document.getElementById("map_decay_c").disabled =
            true;
        document.getElementById("alpha_c").disabled = true;
        document.getElementById("clip_c").disabled = true;
        document.getElementById("use_dropout").disabled =
            true;
        document.getElementById("use_dropout").checked =
            false;
        document.getElementById("use_dropout").onchange();
            //adjusts dropout input = number
        document.getElementById("arch_image").src =
            document.getElementById("arch_image_normal").
            href;
    }
}

function notify_language(e){
    var val = e.value;
    var $toastContent;
    var src_select = document.getElementById("td_src");
    var tgt_select = document.getElementById("td_tgt");
    var dev_src_select = document.getElementById("dd_src");
    var dev_tgt_select = document.getElementById("dd_tgt");
    if(e.id == "td_src"){
        var val2 = tgt_select.value;
        if(( tgt_select.selectedIndex != 0) && (get_extension(
            val) == get_extension(val2))){
            $toastContent = $('<span> Source language
                changed</span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            document.getElementById("dd_src").selectedIndex =
                0;
            document.getElementById("dd_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(( tgt_select.selectedIndex != 0) && (get_name(
            val) != get_name(val2))){
            $toastContent = $('<span> Source dataset changed
                </span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) == "fil"){
            notify_file_length (val, "train");
            $toastContent = $('<span> Source language set to
                Filipino</span>');
        }
        else{
            notify_file_length (val, "train");
            $toastContent = $('<span> Source language set to
                English</span>');
        }
    }
    else if(e.id == "td_tgt"){
        var val2 = src_select.value;
        if( src_select.selectedIndex == 0){
            $toastContent = $('<span> Select a Training source
                language first</span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) == get_extension(val2)){
            $toastContent = $('<span> Invalid target language
                </span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_name(val) != get_name(val2)){
            $toastContent = $('<span> Invalid target dataset
                </span>');
            document.getElementById("td_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) == "fil"){
            notify_file_length (val, "train");
            $toastContent = $('<span> Target language set to
                Filipino</span>');
        }
        else{
            notify_file_length (val, "train");
            $toastContent = $('<span> Target language set to
                English</span>');
```

```
        }
    }
    else if(e.id == "dd_src"){
        var val2 = src_select.value;
        var val3 = dev_tgt_select.value;
        if( src_select.selectedIndex == 0){
            $toastContent = $('<span> Select a Training source
                language first</span>');
            document.getElementById("dd_src").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) != get_extension(val2)){
            $toastContent = $('<span> Invalid Validation
                source language</span>');
            document.getElementById("dd_src").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(( dev_tgt_select.selectedIndex != 0) && (
            get_name(val) != get_name(val3))){
            $toastContent = $('<span> Validation source
                dataset changed</span>');
            document.getElementById("dd_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) == "fil"){
            notify_file_length(val, "dev");
            $toastContent = $('<span> Validation language set
                to Filipino</span>');
        }
        else{
            notify_file_length(val, "dev");
            $toastContent = $('<span> Validation language set
                to English</span>');
        }
    }
    else if(e.id == "dd_tgt"){
        var val2 = tgt_select.value;
        var val3 = dev_src_select.value;
        if( src_select.selectedIndex == 0){
            $toastContent = $('<span> Select a Training source
                language first</span>');
            document.getElementById("dd_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) != get_extension(val2)){
            $toastContent = $('<span> Invalid Validation
                target language</span>');
            document.getElementById("dd_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_name(val) != get_name(val3)){
            $toastContent = $('<span> Invalid Validation
                target dataset</span>');
            document.getElementById("dd_tgt").selectedIndex =
                0;
            $('select').material_select(); //Updates Material
                Select
        }
        else if(get_extension(val) == "fil"){
            notify_file_length(val, "dev");
            $toastContent = $('<span> Validation language set
                to Filipino</span>');
        }
        else{
            notify_file_length(val, "dev");
            $toastContent = $('<span> Validation language set
                to English</span>');
        }
    }
    Materialize.toast($toastContent, 5000);
}

function get_extension(name){
    return name.slice((name.lastIndexOf(".") - 1 >>> 0) + 2);
}

function get_name(name){
    return name.substr(0, name.lastIndexOf('.'));
}

function add_defaults(e){
    if(e.checked == true){
        document.getElementById("optimizer").selectedIndex =
            1;

        document.getElementById("dim_word").value = 512;
        document.getElementById("dim_word").focus(); //lets
            materialize update the object

        document.getElementById("dim").value = 1000;
        document.getElementById("dim").focus();

        document.getElementById("n_words_src").value = 20000;
        document.getElementById("n_words_src").focus();

        document.getElementById("n_words").value = 20000;
        document.getElementById("n_words").focus();

        document.getElementById("lrate").value = 0.0001;
        document.getElementById("lrate").focus();

        document.getElementById("batch_size").value = 80;
        document.getElementById("batch_size").focus();

        document.getElementById("valid_batch_size").value =
            80;
        document.getElementById("valid_batch_size").focus();

        document.getElementById("dispFreq").selectedIndex =
            1;

        document.getElementById("validFreq").selectedIndex =
            1;

        document.getElementById("saveFreq").selectedIndex =
            1;

        document.getElementById("sampleFreq").selectedIndex
            = 1;

        document.getElementById("finish_after").value =
            10000000;
        document.getElementById("finish_after").focus();

        document.getElementById("max_epochs").value = 5000;
        document.getElementById("max_epochs").focus();

        if(document.getElementById("decay_c").disabled ==
            true){
            document.getElementById("decay_c").disabled =
                false;
            document.getElementById("decay_c").value = 0;
            document.getElementById("decay_c").focus();
            document.getElementById("decay_c").blur();
            document.getElementById("decay_c").disabled =
                true;

            document.getElementById("map_decay_c").disabled
                = false;
            document.getElementById("map_decay_c").value =
                0;
            document.getElementById("map_decay_c").focus();
            document.getElementById("map_decay_c").blur();
            document.getElementById("map_decay_c").disabled
                = true;

            document.getElementById("alpha_c").disabled =
                false;
            document.getElementById("alpha_c").value = 0;
            document.getElementById("alpha_c").focus();
            document.getElementById("alpha_c").blur();
            document.getElementById("alpha_c").disabled =
                true;

            document.getElementById("clip_c").disabled = false;
            document.getElementById("clip_c").value = 1.0;
            document.getElementById("clip_c").focus();
            document.getElementById("clip_c").blur();
            document.getElementById("clip_c").disabled = true;
        }
        else{
            document.getElementById("decay_c").value = 0;
            document.getElementById("decay_c").focus();

            document.getElementById("map_decay_c").value =
                0;
            document.getElementById("map_decay_c").focus();

            document.getElementById("alpha_c").value = 0;
            document.getElementById("alpha_c").focus();

            document.getElementById("clip_c").value = 1.0;
            document.getElementById("clip_c").focus();
        }
```

```
        document.getElementById("use_dropout").checked =
            true;
        document.getElementById("use_dropout").onchange();

        document.getElementById("dropout_embedding").value
            = 0.2;
        document.getElementById("dropout_embedding").focus()
            ;

        document.getElementById("dropout_hidden").value =
            0.2;
        document.getElementById("dropout_hidden").focus();

        document.getElementById("dropout_source").value = 0;
        document.getElementById("dropout_source").focus();

        document.getElementById("dropout_target").value = 0;
        document.getElementById("dropout_target").focus();
        document.getElementById("dropout_target").blur();

        document.getElementById("use_dropout").checked =
            false;
        document.getElementById("use_dropout").onchange();

        $('select').material_select(); //Updates Material
            Select
        document.getElementById("suggest_defaults").focus();
    }
    else{

        document.getElementById("optimizer").selectedIndex =
            0;

        document.getElementById("dim_word").value = "";
        document.getElementById("dim_word").focus(); //lets
            materialize update the object

        document.getElementById("dim").value = "";
        document.getElementById("dim").focus();

        document.getElementById("n_words_src").value = "";
        document.getElementById("n_words_src").focus();

        document.getElementById("n_words").value = "";
        document.getElementById("n_words").focus();

        document.getElementById("lrate").value = "";
        document.getElementById("lrate").focus();

        document.getElementById("batch_size").value = "";
        document.getElementById("batch_size").focus();

        document.getElementById("valid_batch_size").value =
            "";
        document.getElementById("valid_batch_size").focus();

        document.getElementById("dispFreq").selectedIndex =
            0;

        document.getElementById("validFreq").selectedIndex =
            0;

        document.getElementById("saveFreq").selectedIndex =
            0;

        document.getElementById("sampleFreq").selectedIndex
            = 0;

        document.getElementById("finish_after").value = "";
        document.getElementById("finish_after").focus();

        document.getElementById("max_epochs").value = "";
        document.getElementById("max_epochs").focus();

        if(document.getElementById("decay_c").disabled ==
            true){
            document.getElementById("decay_c").disabled =
                false;
            document.getElementById("decay_c").value = "";
            document.getElementById("decay_c").focus();
            document.getElementById("decay_c").blur();
            document.getElementById("decay_c").disabled =
                true;

            document.getElementById("map_decay_c").disabled
                = false;
            document.getElementById("map_decay_c").value =
                "";
            document.getElementById("map_decay_c").focus();
            document.getElementById("map_decay_c").blur();
            document.getElementById("map_decay_c").disabled
                = true;

            document.getElementById("alpha_c").disabled =
                false;
            document.getElementById("alpha_c").value = "";
            document.getElementById("alpha_c").focus();
            document.getElementById("alpha_c").blur();
            document.getElementById("alpha_c").disabled =
                true;

            document.getElementById("clip_c").disabled = false;
            document.getElementById("clip_c").value = "";
            document.getElementById("clip_c").focus();
            document.getElementById("clip_c").blur();
            document.getElementById("clip_c").disabled = true;
        }
        else{
            document.getElementById("decay_c").value = "";
            document.getElementById("decay_c").focus();

            document.getElementById("map_decay_c").value =
                "";
            document.getElementById("map_decay_c").focus();

            document.getElementById("alpha_c").value = "";
            document.getElementById("alpha_c").focus();

            document.getElementById("clip_c").value = "";
            document.getElementById("clip_c").focus();
        }

        document.getElementById("use_dropout").checked =
            true;
        document.getElementById("use_dropout").onchange();

        document.getElementById("dropout_embedding").value
            = "";
        document.getElementById("dropout_embedding").focus()
            ;

        document.getElementById("dropout_hidden").value =
            "";
        document.getElementById("dropout_hidden").focus();

        document.getElementById("dropout_source").value = "";
        document.getElementById("dropout_source").focus();

        document.getElementById("dropout_target").value = "";
        document.getElementById("dropout_target").focus();
        document.getElementById("dropout_target").blur();

        document.getElementById("use_dropout").checked =
            false;
        document.getElementById("use_dropout").onchange();

        $('select').material_select(); //Updates Material
            Select
        document.getElementById("suggest_defaults").focus();
    }
}
```

File: neuro-mt\static\style.css

```
ul.dropdown-content.select-dropdown li span {
    color: #0091ea; /* no need for !important :) */
}

.switch label input[type=checkbox]:checked+.lever {
  background-color: #0091ea;
}

.switch label input[type=checkbox]:checked+.lever:after {
  background-color: #fff;
}

/* label color */
.input-field label {
 color: #0091ea;
}
/* label focus color */
.input-field input[type=number]:focus + label {
 color: #0091ea;
}
/* label underline focus color */
.input-field input[type=number]:focus {
 border-bottom: 1px solid #0091ea;
 box-shadow: 0 1px 0 0 #0091ea;
}
/* valid color */
.input-field input[type=number].valid {
 border-bottom: 1px solid #ffd740;
 box-shadow: 0 1px 0 0 #ffd740;
```

```
}
/* invalid color */
.input-field input[type=number].invalid {
 border-bottom: 1px solid #d50000;
 box-shadow: 0 1px 0 0 #d50000;
}
/* icon prefix focus color */
.input-field .prefix.active {
 color: #0091ea;
}


/* label focus color */
.input-field input[type=text]:focus + label {
 color: #0091ea;
}
/* label underline focus color */
.input-field input[type=text]:focus {
 border-bottom: 1px solid #0091ea;
 box-shadow: 0 1px 0 0 #0091ea;
}
/* valid color */
.input-field input[type=text].valid {
 border-bottom: 1px solid #ffd740;
 box-shadow: 0 1px 0 0 #ffd740;
}
/* invalid color */
.input-field input[type=text].invalid {
 border-bottom: 1px solid #d50000;
 box-shadow: 0 1px 0 0 #d50000;
}

/* label focus color */
.input-field textarea:focus + label {
 color: #0091ea;
}

.input-field textarea:focus:not([readonly]) + label {
 color: #0091ea;
}

.input-field textarea:focus:not([readonly]) {
 border-bottom: 1px solid #0091ea;
 box-shadow: 0 1px 0 0 #0091ea;
}
/* label underline focus color */
.input-field textarea:focus {
 border-bottom: 1px solid #0091ea;
 box-shadow: 0 1px 0 0 #0091ea;
}
/* valid color */
.input-field textarea.valid {
 border-bottom: 1px solid #ffd740;
 box-shadow: 0 1px 0 0 #ffd740;
}
/* invalid color */
.input-field textarea.invalid {
 border-bottom: 1px solid #d50000;
 box-shadow: 0 1px 0 0 #d50000;
}

/*material icons*/
/* fallback */
@font-face {
  font-family: 'Material Icons';
  font-style: normal;
  font-weight: 400;
  src: local('Material Icons'), local('MaterialIcons-Regular'),
       url(materialize/fonts/material-icons/material-icons.
       woff2) format('woff2');
}

.material-icons {
  font-family: 'Material Icons';
  font-weight: normal;
  font-style: normal;
  font-size: 24px;
  line-height: 1;
  letter-spacing: normal;
  text-transform: none;
  display: inline-block;
  white-space: nowrap;
  word-wrap: normal;
  direction: ltr;

  font-feature-settings: 'liga'; /* !!! make sure this is set
       !!! */
  -webkit-font-feature-settings: 'liga';
  -webkit-font-smoothing: antialiased;
}
```

File: neuro-mt\static\style_bootstrap.css

```
body
{
        background-color : #F9F9F9;
    min-height: 100vh;
}

.btn-align {
    padding: 6px 12px;
    line-height: 1.42857143;
    vertical-align: middle;
}
```

File: neuro-mt\static\style_materialize.css

```
body {
    display: flex;
    min-height: 100vh;
    flex-direction: column;
}

h1 {
    font-size: 3.5rem;
}

main {
  flex: 1 0 auto;
}

ul.dropdown-content.select-dropdown li span {
    color: #0091ea; /* no need for !important :) */
}

div.carousel.carousel-slider {
    height: 500px;
}

.table-of-contents a.active{
    border-left: 2px solid #0091ea;
}

.table-of-contents a:hover{
    border-left: 1px solid #0091ea;
}

/*material icons*/
/* fallback */
@font-face {
  font-family: 'Material Icons';
  font-style: normal;
  font-weight: 400;
  src: local('Material Icons'), local('MaterialIcons-Regular'),
       url(materialize/fonts/material-icons/material-icons.
       woff2) format('woff2');
}

.material-icons {
  font-family: 'Material Icons';
  font-weight: normal;
  font-style: normal;
  font-size: 24px;
  line-height: 1;
  letter-spacing: normal;
  text-transform: none;
  display: inline-block;
  white-space: nowrap;
  word-wrap: normal;
  direction: ltr;

  font-feature-settings: 'liga'; /* !!! make sure this is set
       !!! */
  -webkit-font-feature-settings: 'liga';
  -webkit-font-smoothing: antialiased;
}
```

File: neuro-mt\templates\about.html

```
{% extends "base.html" %}

{% block title %}Neuro MT{% endblock %}

{% block content %}
<div class="navbar-fixed">
<nav class = "light-blue accent-4">
<div class="container">
    <div class="nav-wrapper">
      <a href="{{ url_for('index') }}" class="brand-logo">
      <span class="white-text">Neuro </span>
      <span class="amber-text text-accent-2">MT</span
           ></a>
    </div>
```

```
    <ul id="nav-mobile" class="right">
        <li><a href="{{ url_for('api') }}">API</a></li>
        <li><a href="{{ url_for('about') }}">About</a></li
            >
    </ul>
</div>
</nav>
</div>

<div class="container">
    <div class="row">
        <img src="{{url_for('static', filename='NeuroMT Logo.
            png')}}" style="height: 12rem; display: block;
            margin: 0 auto;"/>
        <h3 class="col s12 light center header light-blue-text
             text-accent-4">
        Neuro MT: A Learning Machine Translation Web Service
            , is made possible by</h3>
    </div>
    <div class="row">
        <div class="col s4">
            <div class="center promo promo-example">
                <img src="{{url_for('static', filename='
                    UP_Manila.png')}}" style="height: 6rem
                    ;"/>
                <p class="promo-caption">UP Manila</p>
                <p class="light center">University of the
                    Philippines Manila is a coeducational and
                     public research university in the
                    Philippines.</p>
            </div>
        </div>
        <div class="col s4">
            <div class="center promo promo-example">
                <img src="{{url_for('static', filename='pcari.
                    jpg')}}" style="height: 6rem;"/>
                <p class="promo-caption">PCARI Project</p
                    >
                <p class="light center">Is a new approach to
                    enhance the skills and expertise of faculty
                     and staff of Philippine universities and
                     colleges.
                    They provided the initial dataset for this
                         project.</p>
            </div>
        </div>
        <div class="col s4">
            <div class="center promo promo-example">
                <img src="{{url_for('static', filename='github.
                    png')}}" style="height: 6rem;"/>
                <p class="promo-caption">Nematus</p>
                <p class="light center">An Attention-based
                    encoder-decoder model for neural
                    machine translation based on the dl4mt-
                    tutorial by Kyunghyun Cho et al.</p>
            </div>
        </div>
    </div>
</div>

<footer class="page-footer light-blue accent-4">
    <div class="container">
        <div class="row">
            <div class="col l6 s12">
            <h5 class="white-text">Neuro <span class="
                amber-text text-accent-2">MT</span></
                h5>
            <p class="grey-text text-lighten-4">
                An application with a built-in REST API and
                    trainer. It aims to provide developers
                a way to easily serve and update the translator
                    of their preference.
            </p>
            </div>
            <div class="col l4 offset-l2 s12">
            <h5 class="white-text">Links</h5>
            <ul>
                <li><a class="grey-text text-lighten-3" href
                    ="http://dpsm.cas.upm.edu.ph/">UP
                    Manila DPSM</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="http://pcariofficial.blogspot.com/">
                    PCARI Project</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="https://github.com/rsennrich/nematus
                    ">Nematus</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="{{ url_for('translateapp') }}">Neuro
                    Translate</a></li>
            </ul>
            </div>
        </div>
    </div>
    <div class="footer-copyright light-blue darken-3">
        <div class="container">
            2017 Neuro MT, All rights reserved.
            <a class="grey-text text-lighten-4 right" href="https
                ://ph.linkedin.com/in/alfred-tacorda-a932a611b
                ">Alfred Tacorda</a>
        </div>
    </div>
</footer>
{% endblock %}

{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('.static', filename='
    style_materialize.css')}}">
{% endblock %}

{% block scripts %}
{{ super() }}
<script src="{{url_for('.static', filename='scripts_materialize.
    js')}}"></script>
{% endblock %}

File: neuro-mt\templates\api.html

{% extends "base.html" %}

{% block title %}Neuro MT{% endblock %}

{% block content %}
<div class="navbar-fixed">
<nav class = "light-blue accent-4">
<div class="container">
    <div class="nav-wrapper">
      <a href="{{ url_for('index') }}" class="brand-logo">
      <span class="white-text">Neuro </span>
      <span class="amber-text text-accent-2">MT</span
          ></a>
    </div>

    <ul id="nav-mobile" class="right">
        <li><a href="{{ url_for('api') }}">API</a></li>
        <li><a href="{{ url_for('about') }}">About</a></li
            >
    </ul>
</div>
</nav>
</div>

<div class="container">
    <div class = "row">
        <div class = "col s12 m9 l10">
            <div id="translationRequest" class="section
                scrollspy">
                <h4 class="light-blue-text text-accent-4">
                    Translation Request</h4>
                <p style="font-size: 1.2rem;">To get a
                    translation, your app must first make a
                    translation request.</p>
                <div class="row">
                    <div class="col s6">
                        <h5>Request</h5>
                        <div class="card small">
                            <div class="card-content">
                                <span class="card-title light-
                                    blue-text text-accent
                                    -4">POST: /translate</
                                    span>
                            </div>
                            <div class="card-content">
                                <p>{</p>
                                <p>    "source": " fil",</p>
                                <p>    "target": "eng",</p>
                                <p>    "input": "Ito ay isang
                                    pangungusap."</p>
                                <p>}</p>
                            </div>
                        </div>
                    </div>
                    <div class="col s6">
                        <h5>Result</h5>
                        <div class="card small">
                            <div class="card-content">
                                <span class="card-title light-
                                    blue-text text-accent
                                    -4">Status: 202
                                    ACCEPTED</span>
                            </div>
                            <div class="card-content">
                                <p>{</p>
```

```
                    <p>    "location": "/translate
                           /< request-id >"</p>
                    <p>}</p>
                </div>
            </div>
        </div>
    </div>
</div>
<div id="getTranslation" class="section scrollspy
    ">
    <h4 class="light-blue-text text-accent-4">
        Get Translation</h4>
    <p style="font-size: 1.2rem;">
        If the returned result of the translation
            request is 202, it means that your
            request is successfully added to our
            translation queue.
        To get the translation, perform long-
            polling in the returned location.
    </p>
    <div class="row">
        <div class="col s6">
            <h5>Request</h5>
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">GET: /translate/<
                        request-id ></span>
                </div>
                <div class="card-content">
                    <p>Situation: Your request is
                        successfully translated.<
                        p>
                </div>
            </div>
        </div>
        <div class="col s6">
            <h5>Result</h5>
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">Status: 200 OK</
                        span>
                </div>
                <div class="card-content">
                    <p>{</p>
                    <p>    "state": "SUCCESS
                        ",</p>
                    <p>    "translation": "This is
                        a sentence."</p>
                    <p>}</p>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col s6">
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">GET: /translate/<
                        request-id ></span>
                </div>
                <div class="card-content">
                    <p>Situation: Your request is
                         still not processed by our
                         model.</p>
                </div>
            </div>
        </div>
        <div class="col s6">
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">Status: 200 OK</
                        span>
                </div>
                <div class="card-content">
                    <p>{</p>
                    <p>    "state": "PENDING
                        ",</p>
                    <p>    "translation": ""</p>
                    <p>}</p>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col s6">
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">GET: /translate/<
                        request-id ></span>
                </div>
                <div class="card-content">
                    <p>Situation: Your request is
                        being translated by our
                        model.</p>
                </div>
            </div>
        </div>
        <div class="col s6">
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">Status: 200 OK</
                        span>
                </div>
                <div class="card-content">
                    <p>{</p>
                    <p>    "state": "
                        TRANSLATING",</p>
                    <p>    "translation": ""</p>
                    <p>}</p>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col s6">
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">GET: /translate/<
                        request-id ></span>
                </div>
                <div class="card-content">
                    <p>Situation: Sudden
                        interruption of the service
                        .</p>
                </div>
            </div>
        </div>
        <div class="col s6">
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">Status: 200 OK</
                        span>
                </div>
                <div class="card-content">
                    <p>{</p>
                    <p>    "state": "FAILED",</p
                        >
                    <p>    "translation": ""</p>
                    <p>}</p>
                </div>
            </div>
        </div>
    </div>
</div>
<div id="suggestions" class="section scrollspy">
    <h4 class="light-blue-text text-accent-4">
        Suggestions</h4>
    <p style="font-size: 1.2rem;">You can allow
        your users to suggest translations and
        improve our model.</p>
    <div class="row">
        <div class="col s6">
            <h5>Request</h5>
            <div class="card small">
                <div class="card-content">
                    <span class="card-title light-
                        blue-text text-accent
                        -4">POST: /suggest</
                        span>
                </div>
                <div class="card-content">
                    <p>{</p>
                    <p>    "eng": "This is a
                        suggested sentence.",</p
                        >
                    <p>    "fil": "Ito ay isang
                        isinuhestyong
                        pangungusap."</p>
```

```
                                <p>}</p>
                            </div>
                        </div>
                    </div>
                    <div class="col s6">
                        <h5>Result</h5>
                        <div class="card small">
                            <div class="card-content">
                                <span class="card-title light-
                                    blue-text text-accent
                                    -4">Status: 200 OK</
                                    span>
                            </div>
                            <div class="card-content">
                                <p>Note: </p>
                                <p>The server will filter out
                                    the suggestion if both
                                    fields are empty.
                                    This means that you can
                                    suggest the input of
                                    the user or the
                                    output of the server
                                    .</p>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="col hide-on-small-only m3 l2">
            <div class="toc-wrapper">
                <ul class="section table-of-contents">
                    <li><a href="#translationRequest">
                        Translation Request</a></li>
                    <li><a href="#getTranslation">Get
                        Translation</a></li>
                    <li><a href="#suggestions">Suggestions
                        </a></li>
                </ul>
            </div>
        </div>
    </div>
</div>
<footer class="page-footer light-blue accent-4">
    <div class="container">
        <div class="row">
            <div class="col l6 s12">
            <h5 class="white-text">Neuro <span class="
                amber-text text-accent-2">MT</span></
                h5>
            <p class="grey-text text-lighten-4">
                An application with a built-in REST API and
                    trainer. It aims to provide developers
                a way to easily serve and update the translator
                    of their preference.
            </p>
            </div>
            <div class="col l4 offset-l2 s12">
            <h5 class="white-text">Links</h5>
            <ul>
                <li><a class="grey-text text-lighten-3" href
                    ="http://dpsm.cas.upm.edu.ph/">UP
                    Manila DPSM</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="http://pcariofficial.blogspot.com/">
                    PCARI Project</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="https://github.com/rsennrich/nematus
                    ">Nematus</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="{{ url_for('translateapp') }}">Neuro
                    Translate</a></li>
            </ul>
            </div>
        </div>
    </div>
    <div class="footer-copyright light-blue darken-3">
        <div class="container">
            2017 Neuro MT, All rights reserved.
        <a class="grey-text text-lighten-4 right" href="https
            ://ph.linkedin.com/in/alfred-tacorda-a932a611b
            ">Alfred Tacorda</a>
        </div>
    </div>
</footer>
{% endblock %}

{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('. static ', filename='
    style_materialize .css ')}}">
{% endblock %}
```

```
{% block scripts %}
{{ super() }}
<script src="{{url_for('. static ', filename='scripts_materialize .
    js ')}}"></script>
{% endblock %}
```

File: neuro-mt\templates\base.html

```
<!-- Code copied from Flask-Materialize. Availible at https://
    github.com/HellerCommaA/flask-materialize -->
{% macro row() %}<div class="row">{% endmacro %}

{% macro container() %}<div class="container">{% endmacro
    %}

{% macro col(cols=[]) %}<div class="col {{ cols|join(' ')
    }}">{% endmacro %}

{% macro enddiv() %}</div>{% endmacro %}

{% block doc -%}
<!DOCTYPE html>
<html{% block html_attribs %}{% endblock html_attribs %}>
{%- block html %}
  <head>
    {%- block head %}
    <title>{% block title %}{% endblock title %}</title>

    {%- block metas %}
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
    {%- endblock metas %}

    {%- block styles %}
    <!-- Materialize. Availible at https://github.com/Dogfalo/
        materialize -->
    <link type="text/css" rel="stylesheet" href="{{url_for('.
        static ', filename='materialize/css/materialize.min.css
        ')}}"  media="screen,projection"/>
    {%- endblock styles %}
    {%- endblock head %}
  </head>
  <body{% block body_attribs %}{% endblock body_attribs
      %}>
    {% block body -%}
    {% block navbar %}
    {%- endblock navbar %}
    {% block content -%}
    {%- endblock content %}

    {% block scripts %}
    <script type="text/javascript" src="{{url_for('. static ',
        filename='jquery-2.1.1.min.js', cdn='jquery')}}"></
        script>
    <script type="text/javascript" src="{{url_for('. static ',
        filename='materialize/js/materialize.js', cdn='
        material')}}"></script>
    {%- endblock scripts %}
    {% block footer %}
    {%- endblock footer %}
    {%- endblock body %}
  </body>
{%- endblock html %}
</html>
{% endblock doc -%}
```

File: neuro-mt\templates\bootstrap_base.html

```
{# ``base.html`` is the template all our other templates derive
    from. While
   Flask-Bootstrap ships with its own base, it is good form to
       create a custom
   one for our app, as it allows customizing some aspects.

   Deriving from bootstap/base.html gives us a basic page
       scaffoling.

   You can find additional information about template
       inheritance at

   http://jinja.pocoo.org/docs/templates/#template-
       inheritance
#}
{% extends "bootstrap/base.html" %}

{# We also set a default title, usually because we might forget
    to set one.
   In our sample app, we will most likely just opt not to
       change it #}
{% block title %}Neuro Translate{% endblock %}
```

```
{# While we are at it, we also enable fixes for legacy browsers.
        First we
   import the necessary macros: #}
{# import "bootstrap/fixes.html" as fixes #}

{# Then, inside the head block, we apply these. To not replace
       the header,
   ``super()`` is used: #}
{% block head %}
{{super()}}

{#- Docs: http://pythonhosted.org/Flask-Bootstrap/macros.
       html#fixes
   The sample application already contains the required static
          files . #}
{#{fixes.ie8()}#}
{% endblock %}

{# Adding our own CSS files is also done here. Check the
       documentation at
   http://pythonhosted.org/Flask-Bootstrap/basic-usage.html
          #available-blocks
   for an overview. #}
{% block styles %}
{{super()}}
<link rel="stylesheet"
      href="{{url_for('.static', filename='style_bootstrap.css')
            }}">
{% endblock %}

{% block scripts %}
{{super()}}
<script src="{{url_for('.static', filename='scripts_bootstrap.js
      ')}}"></script>
{% endblock %}

{# Finally, round things out with navigation #}
{% block navbar %}
        <nav class="navbar navbar-default" style="
        background-color: #3F51B5;
        border: none;
        box-shadow: 0px 2px 1px #FFD700;
        border-radius: 0px;
        ">
          <div class="container-fluid">
                <div class="navbar-header">
                  <a class="navbar-brand" href="{{url_for('
                        translateapp')}}" style="color: #
                        FFFFFF">Neuro <span style="color:
                        #FFD700">Translate</span></a>
                </div>
          </div>
        </nav>
{% endblock %}

File: neuro-mt\templates\data.html

{% extends "base.html" %}

{% block title %}Neuro MT{% endblock %}

{% block content %}
<div class="navbar-fixed">
<nav class = "light-blue accent-4">
<div class="container">
    <div class="nav-wrapper">
      <a href="{{ url_for('index') }}" class="brand-logo">
      <span class="white-text">Neuro </span>
      <span class="amber-text text-accent-2">MT</span
            ></a>
    </div>

    <ul id="nav-mobile" class="right">
        <li><a href="{{url_for('train')}}">Train</a></li>
        <li><a href="{{url_for('test')}}">Test</a></li>
        <li class="active"><a href="{{url_for('data')}}">
            Data</a></li>
    </ul>
</div>
</nav>
</div>
<div class="container">
    <div class="section">
        <div class="row">
            <div class="col s8">
               <div class="card small sticky-action">
                      <div class="card-content light-blue
                            accent-4 white-text">
                        <p class="card-stats-title" style="font
                              -size: 1.5rem"><i class="small
                              material-icons" style="vertical-
                              align:top !important;">note_add</
                              i> Update Dataset</p>
</div>
<div class="card-tabs">
  <ul class="tabs tabs-fixed-width">
    <li class="tab"><a href="#train"
          class="active"><span class="
          light-blue-text text-accent
          -4">Training</span></a></
          li>
    <li class="tab"><a class="" href
          ="#test"><span class="light-
          blue-text text-accent-4">
          Testing</span></a></li>
    <li class="tab"><a class="" href
          ="#dev"><span class="light-
          blue-text text-accent-4">
          Validation</span></a></li>
  <div class="indicator amber accent-2"
        style="right: 239px; left: 0px; z-
        index:1;"></div></ul>
</div>
<div class="card-content light-blue-
      text text-accent-4" style="
      overflow:visible;">
  <div id="train" class="row">
      <div class="input-field col s12
            light-blue-text text-accent
            -4">
          <select id = "tnd_append">
            <option value="" disabled
                  selected >Select a
                  Training Dataset</
                  option>
            {% for td in ndlist %}
            <option value='{{td}}' >{{
                  td}}</option>
            {% endfor %}
          </select>
          <label>Training Dataset</
                label>
      </div>

      <button class="btn-floating waves
            -effect waves-light light-
            blue accent-4 btn-large right
            " id="addTrain" href="{{
            url_for('data_add')}}">
      <i class="material-icons">add</i
            >
      </button>
  </div>
  <div id="test" class="row">
      <div class="input-field col s12
            light-blue-text text-accent
            -4">
          <select id = "ttd_append">
            <option value="" disabled
                  selected >Select a
                  Testing Dataset</
                  option>
            {% for td in ndlist %}
            <option value='{{td}}' >{{
                  td}}</option>
            {% endfor %}
          </select>
          <label>Testing Dataset</label
                >
      </div>
      <button class="btn-floating waves
            -effect waves-light light-
            blue accent-4 btn-large right
            " id="addTest" href="{{
            url_for('data_add')}}">
      <i class="material-icons">add</i
            >
      </button>
  </div>
  <div id="dev" class="row">
      <div class="input-field col s12
            light-blue-text text-accent
            -4">
          <select id = "vdd_append">
            <option value="" disabled
                  selected >Select a
                  Validation Dataset</
                  option>
            {% for td in ndlist %}
            <option value='{{td}}' >{{
                  td}}</option>
            {% endfor %}
          </select>
```

```
                        <label>Validation Dataset</
                            label>
                    </div>
                    <button class="btn-floating waves
                        -effect waves-light light-
                        blue accent-4 btn-large right
                        " id="addDev" href="{{
                        url_for('data_add')}}">
                    <i class="material-icons">add</i
                        >
                    </button>
                </div>
              </div>
        </div>
    </div>
     <div class="col s4">
         <div class="card center-align">
             <div class="card-content light-blue accent
                 -4 white-text">
                 <p class="card-stats-title" style="
                     font-size: 1.3rem"><i class="
                     small material-icons " style="
                     vertical-align:middle !important
                     ;">chat_bubble_outline</i> New
                     Suggestions</p>
                 <h4 class="card-stats-number"><
                     span class="amber-text text-
                     accent-2" id="suggestions_count
                     ">{{suggestions_count}}</span
                     ></h4>
                 </p>
             </div>
             <div class="card-action white">
                 <button class="btn waves-effect waves
                     -light white light-blue-text text
                     -accent-4 btn" href="{{url_for('
                     data_save')}}" id="save" style="
                     margin: 0; box-shadow: none;">
                     Save Suggestions</button>
             </div>
         </div>
    </div>
</div>
<div class="row">
    <div class="col s8">
        <div class="card small sticky-action">
                <div class="card-content light-blue
                     accent-4 white-text">
                  <p class="card-stats-title" style="font
                      -size: 1.5rem"><i class="small
                      material-icons" style="vertical-
                      align:top !important;">
                      remove_circle</i> Remove Dataset
                      </p>
                </div>
                <div class="card-tabs">
                  <ul class="tabs tabs-fixed-width">
                    <li class="tab"><a href="#
                        trainGet" class="active"><
                        span class="light-blue-text
                        text-accent-4">Training</
                        span></a></li>
                    <li class="tab"><a class="" href
                        ="#testGet"><span class="
                        light-blue-text text-accent
                        -4">Testing</span></a></li
                        >
                    <li class="tab"><a class="" href
                        ="#devGet"><span class="
                        light-blue-text text-accent
                        -4">Validation</span></a
                        ></li>
                  <div class="indicator amber accent-2"
                        style="right: 239px; left: 0px; z-
                        index:1;"></div></ul>
                </div>
                <div class="card-content light-blue-
                     text text-accent-4" style="
                     overflow:visible;">
                  <div id="trainGet" class="row">
                      <div class="input-field col s12
                          light-blue-text text-accent
                          -4">
                        <select id = "tnd_remove">
                          <option value="" disabled
                              selected >Select a
                              Training Dataset</
                              option>
                          {% for td in tnlist %}
                          <option value='{{td}}' >{{
                              td}}</option>
                          {% endfor %}
                        </select>
                        <label>Training Dataset</
                            label>
                      </div>
                      <button class="btn-floating waves
                          -effect waves-light red
                          darken-1 btn-large right" id
                          ="removeTrain" href="{{
                          url_for('data_remove')}}">
                      <i class="material-icons">remove
                          </i>
                      </button>
                  </div>
                  <div id="testGet" class="row">
                      <div class="input-field col s12
                          light-blue-text text-accent
                          -4">
                        <select id = "ttd_remove">
                          <option value="" disabled
                              selected >Select a
                              Testing Dataset</
                              option>
                          {% for td in ttlist %}
                          <option value='{{td}}' >{{
                              td}}</option>
                          {% endfor %}
                        </select>
                        <label>Testing Dataset</label
                            >
                      </div>
                      <button class="btn-floating waves
                          -effect waves-light red
                          darken-1 btn-large right" id
                          ="removeTest" href="{{
                          url_for('data_remove')}}">
                      <i class="material-icons">remove
                          </i>
                  </div>
                  <div id="devGet" class="row">
                      <div class="input-field col s12
                          light-blue-text text-accent
                          -4">
                        <select id = "vdd_remove">
                          <option value="" disabled
                              selected >Select a
                              Validation Dataset</
                              option>
                          {% for td in dlist %}
                          <option value='{{td}}' >{{
                              td}}</option>
                          {% endfor %}
                        </select>
                        <label>Validation Dataset</
                            label>
                      </div>
                      <button class="btn-floating waves
                          -effect waves-light red
                          darken-1 btn-large right" id
                          ="removeDev" href="{{
                          url_for('data_remove')}}">
                      <i class="material-icons">remove
                          </i>
                      </button>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
    </div>
</div>
{% endblock %}

{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('.static', filename='
    style.css')}}">
{% endblock %}

{% block scripts %}
{{ super() }}
<script src="{{url_for('.static', filename='scripts_data.js')
    }}"></script>
{% endblock %}
```

File: neuro-mt\templates\index.html

```
{% extends "base.html" %}

{% block title %}Neuro MT{% endblock %}

{% block content %}
<div class="navbar-fixed">
```

```
<nav class = "light-blue accent-4">
<div class="container">
    <div class="nav-wrapper">
      <a href="{{ url_for('index') }}" class="brand-logo">
      <span class="white-text">Neuro </span>
      <span class="amber-text text-accent-2">MT</span
          ></a>
    </div>

    <ul id="nav-mobile" class="right">
        <li><a href="{{url_for('train')}}">Train</a></li>
        <li><a href="{{url_for('test')}}">Test</a></li>
        <li><a href="{{url_for('data')}}">Data</a></li>
    </ul>
</div>
</nav>
</div>
<div class="container">
    <div class="section">
        <h5><span class = "light-blue-text text-accent
            -4">{{trainName}}</span></h5>
        <div class="row">
            <div class="col s9">
                <embed type="image/svg+xml" src={{
                    graph_data|safe}} style='max-width
                    :100%'/>
            </div>
            <div class="col s3">
                <div class="card center-align">
                    <div class="card-content light-blue accent
                        -4 white-text">
                        <p class="card-stats-title" style="
                            font-size: 1.3rem"><i class="
                            small material-icons " style="
                            vertical-align:middle !important
                            ;">assessment</i>Loss | BLEU
                            </p>
                        <h4 class="card-stats-number"><
                            span class="amber-text text-
                            accent-2" id="suggestions_count
                            ">{{loss}} | {{bleu}}</span></
                            h4>
                        </p>
                    </div>
                    <div class="card-action white">
                    {% if state == "TRAINING" %}
                        <button class="btn waves-effect waves
                            -light white light-blue-text text
                            -accent-4" href="{{url_for('
                            stopTraining')}}" id="
                            stopTraining" style="box-shadow
                            : none;">Stop</button>
                    {% else %}
                        <a href="{{url_for('train')}}" class="
                            light-blue-text text-accent-4"
                            style="margin-right: 0;">Start
                            Training</a>
                    {% endif %}
                    </div>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col s9">
                <embed type="image/svg+xml" src={{
                    graph_data2|safe}} style='max-width
                    :100%'/>
            </div>
        </div>
    </div>
</div>
{% endblock %}


{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('. static ', filename='
    style.css')}}">
{% endblock %}

{% block scripts %}
{{ super() }}
<script src="{{url_for('. static ', filename='scripts_index.js')
    }}"></script>
{% endblock %}

File: neuro-mt\templates\index_rest.html

{% extends "base.html" %}

{% block title %}Neuro MT{% endblock %}

{% block content %}
<div class="navbar-fixed">
<nav class = "light-blue accent-4">
<div class="container">
    <div class="nav-wrapper">
      <a href="{{ url_for('index') }}" class="brand-logo">
      <span class="white-text">Neuro </span>
      <span class="amber-text text-accent-2">MT</span
          ></a>
    </div>

    <ul id="nav-mobile" class="right">
        <li><a href="{{ url_for('api') }}">API</a></li>
        <li><a href="{{ url_for('about') }}">About</a></li
            >
    </ul>
</div>
</nav>
</div>

<div class="carousel carousel-slider center" data-indicators="
    true">
    <div class="carousel-fixed-item center">
        <a class="btn waves-effect white grey-text darken-
            text-2" href="{{ url_for('api') }}">Go to API</
            a>
    </div>
    <div class="carousel-item amber white-text" href="#one
        !">
        <br><br>
        <h1>Machine Translation for Everyone</h1>
        <p class="white-text" style="font-size: 2rem;">
            Uses REST API, a common language of HTTP-
                Based services.
        </p>
    </div>
    <div class="carousel-item blue white-text" href="#two
        !">
        <br><br>
        <h1>A Learning Web Service</h1>
        <p class="white-text" style="font-size: 2rem;">
            Suggestions received by the system are taken as
                data for retraining.
        </p>
    </div>
    <div class="carousel-item white light-blue-text" href="#
        three!">
        <br><br>
        <h1>Bleeding-Edge Artificial Intelligence</h1>
        <p class="light-blue-text" style="font-size: 2rem;">
            The system uses Nematus. An Attention-based
                encoder-decoder model for neural machine
                translation
            based on the dl4mt-tutorial by Kyunghyun Cho et
                al.
        </p>
    </div>
</div>
<footer class="page-footer light-blue accent-4">
    <div class="container">
        <div class="row">
            <div class="col l6 s12">
            <h5 class="white-text">Neuro <span class="
                amber-text text-accent-2">MT</span></
                h5>
            <p class="grey-text text-lighten-4">
                An application with a built-in REST API and
                    trainer. It aims to provide developers
                a way to easily serve and update the translator
                    of their preference.
            </p>
            </div>
            <div class="col l4 offset-l2 s12">
            <h5 class="white-text">Links</h5>
            <ul>
                <li><a class="grey-text text-lighten-3" href
                    ="http://dpsm.cas.upm.edu.ph/">UP
                    Manila DPSM</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="http://pcariofficial.blogspot.com/">
                    PCARI Project</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="https://github.com/rsennrich/nematus
                    ">Nematus</a></li>
                <li><a class="grey-text text-lighten-3" href
                    ="{{ url_for('translateapp') }}">Neuro
                    Translate</a></li>
            </ul>
            </div>
        </div>
    </div>
    <div class="footer-copyright light-blue darken-3">
```

```
        <div class="container">
          2017 Neuro MT, All rights reserved.
        <a class="grey-text text-lighten-4 right" href="https
            ://ph.linkedin.com/in/alfred-tacorda-a932a611b
            ">Alfred Tacorda</a>
        </div>
    </div>
</footer>
{% endblock %}


{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('. static ', filename='
    style_materialize .css ')}}">
{% endblock %}

{% block scripts %}
{{ super() }}
<script src="{{url_for('. static ', filename='scripts_materialize .
    js ')}}"></script>
{% endblock %}

File: neuro-mt\templates\test.html

{% extends "base.html" %}

{% block title %}Neuro MT{% endblock %}

{% block content %}
<div class="navbar-fixed">
<nav class = "light-blue accent-4">
<div class="container">
    <div class="nav-wrapper">
      <a href="{{ url_for('index') }}" class="brand-logo">
      <span class="white-text">Neuro </span>
      <span class="amber-text text-accent-2">MT</span
          ></a>
    </div>

    <ul id="nav-mobile" class="right">
        <li><a href="{{url_for('train')}}">Train</a></li>
        <li class="active"><a href="{{url_for('test')}}">Test
            </a></li>
        <li><a href="{{url_for('data')}}">Data</a></li>
    </ul>
</div>
</nav>
</div>
<div class="container">
    <div class="section">
        <h5><span class = "light-blue-text text-accent-4">
            Select Test Scenario</span></h5>
        <div class="row"> <!-- test_model -->
            <div class="col s6">
              <div class="row">
                <div class="input-field col s12 light-blue-
                    text text-accent-4 tooltipped"
                  data-position="right" data-delay="30" data
                      -tooltip="<name>.<SRCTGT>.npz">
                    <select id = "base_model" onchange="
                        notify_language(this)">
                      <option value="" disabled selected >
                          Select Base Model</option>
                      {% for tm in mlist %}
                      <option value='{{tm}}' >{{tm}}</
                          option>
                      {% endfor %}
                    </select>
                    <label>Base Model</label>
                  </div>

                  <div class="input-field col s12 light-blue-
                      text text-accent-4 tooltipped"
                  data-position="right" data-delay="30" data
                      -tooltip="<name>.<iter/best_bleu>.
                      npz">
                    <select id = "test_model" onchange="
                        notify_language(this)">
                      <option value="" disabled selected >
                          Select a Model</option>
                      {% for tm in mlist %}
                      <option value='{{tm}}' >{{tm}}</
                          option>
                      {% endfor %}
                    </select>
                    <label>Model to be Tested</label>
                  </div>

                  <div class="input-field col s12 light-blue-
                      text text-accent-4 tooltipped"
                  data-position="right" data-delay="30" data
                      -tooltip="Source Language Test Set">
                    <select id = "td_src" onchange="
                        notify_language(this)">
                      <option value="" disabled selected >
                          Select a Source Testing Dataset</
                          option>
                      {% for td in  tlist  %}
                      <option value='{{td}}' >{{td}}</option
                          >
                      {% endfor %}
                    </select>
                    <label>Testing Dataset (Source)</label>
                  </div>

                  <div class="input-field col s12 light-blue-
                      text text-accent-4 tooltipped"
                  data-position="right" data-delay="30" data
                      -tooltip="Target Language Test Set">
                    <select id = "td_tgt" onchange="
                        notify_language(this)">
                      <option value="" disabled selected >
                          Select a Target Testing Dataset</
                          option>
                      {% for td in  tlist  %}
                      <option value='{{td}}' >{{td}}</option
                          >
                      {% endfor %}
                    </select>
                    <label>Testing Dataset (Target)</label>
                  </div>
                </div>
              </div>

              <div class="col s4 offset-s1">
                <div class="card center-align">
                  <div class="card-content light-blue accent
                      -4 white-text">
                    <p class="card-stats-title" style="
                        font-size: 1.5rem"><i class="
                        small material-icons" style="
                        vertical-align:top !important;">
                        assessment</i> BLEU Score</p
                        >
                    <h4 class="card-stats-number"><
                        span class="amber-text text-
                        accent-2" id="bleu_score
                        ">--</span></h4>
                    </p>
                  </div>
                  <div class="card-action white">
                    <button class="btn waves-effect waves
                        -light white light-blue-text text
                        -accent-4 btn" href="{{url_for('
                        testmodel')}}" id="test" style="
                        box-shadow: none;">Test</
                        button>
                    <button class="btn waves-effect waves
                        -light white light-blue-text text
                        -accent-4 btn" href="{{url_for('
                        deployModel')}}" id="deploy"
                        style="box-shadow: none;">
                        Deploy</button>
                  </div>
                </div>
              </div>

              <div class="row">
                <div class="input-field col s5">
                  <textarea id="input" class="materialize-
                      textarea"></textarea>
                  <label for="input">Input</label>
                </div>
                <div class="input-field col s5">
                  <textarea id="translation" class="materialize-
                      textarea" readonly></textarea>
                  <label for="translation">Translation</label>
                </div>
                <div class="col s2">
                  <button href="{{url_for('testtranslate')}}"
                      class="btn waves-effect waves-light light
                      -blue accent-4 btn-medium right" id="
                      testTranslate">Translate
                  </button>
                </div>
              </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}

{% block styles %}
{{super()}}
```

```
<link rel="stylesheet" href="{{url_for('. static ', filename='
      style.css')}}">
{% endblock %}

{% block scripts %}
{{ super() }}
<script src="{{url_for('. static ', filename='scripts_test .js')
      }}"></script>
{% endblock %}

File: neuro-mt\templates\train.html

{% extends "base.html" %}

{% block title %}Neuro MT{% endblock %}

{% block content %}
<div class="navbar-fixed">
<nav class = "light-blue accent-4">
<div class="container">
    <div class="nav-wrapper">

      <a href="{{ url_for('index') }}" class="brand-logo">
      <span class="white-text">Neuro </span>
      <span class="amber-text text-accent-2">MT</span
          ></a>
    </div>

    <ul id="nav-mobile" class="right">
        <li class="active"><a href="{{url_for('train')}}">
            Train</a></li>
        <li><a href="{{url_for('test')}}">Test</a></li>
        <li><a href="{{url_for('data')}}">Data</a></li>
    </ul>
</div>
</nav>
</div>

<div class="container">
    <div class="section">
        <h5><span class = "light-blue-text text-accent-4">
            Training</span></h5>
        <p>
            The system uses nematus, an Attention-based
                encoder-decoder model for neural machine
                translation
            based on the dl4mt-tutorial by Kyunghyun Cho et
                al. The system also implements Byte Pair
                Encoding,
            a technique presented by Sennrich et al. to acheive
                an open vocabulary Neural Machine
                Translation.
        </p>
    </div>
    <ul class="collapsible" data-collapsible="expandable">
    <li>
    <div class="collapsible-header"><i class="material-icons
        ">assignment</i>Step 1: Prepare Dataset</div>
    <div class="collapsible-body">
        <h5><span class = "light-blue-text text-accent-4">
            Enter Training Name</span></h5>
        <div class="row"> <!-- dim -->
          <div class="input-field col s6 light-blue-text text-
              accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
              ="Name to be used by the whole training">
            <input id="name" type="text" class="validate
                light-blue-text text-accent-4">
            <label for="name" data-error="wrong" data-
                success="right">Training Name</label>
          </div>
        </div>

        <h5><span class = "light-blue-text text-accent-4">
            Prepare Dataset</span></h5>

        <a id="getFileLength" href="{{url_for('getFileLength')
            }}" style="display: none;"></a>

        <div class="row"> <!-- td_src -->
          <div class="input-field col s6 light-blue-text text-
              accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
              ="Training Dataset for the Source Language">
            <select id = "td_src" onchange="notify_language(
                this)">
              <option value="" disabled selected >Select a
                  Source Training Dataset</option>
            {% for td in  tlist  %}
            <option value='{{td}}' >{{td}}</option>
            {% endfor %}
            </select>
            <label>Training Dataset (Source)</label>
          </div>
        </div>

        <div class="row"> <!-- td_tgt -->
          <div class="input-field col s6 light-blue-text text-
              accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
              ="Training Dataset for the Target Language">
            <select id = "td_tgt" onchange="notify_language(
                this)">
              <option value="" disabled selected >Select a
                  Target Training Dataset</option>
            {% for td in  tlist  %}
            <option value='{{td}}' >{{td}}</option>
            {% endfor %}
            </select>
            <label>Training Dataset (Target)</label>
          </div>
        </div>

        <div class="row"> <!-- dd_src -->
          <div class="input-field col s6 light-blue-text text-
              accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
              ="Validation or Development Dataset for the
              Source Language">
            <select id = "dd_src" onchange="notify_language(
                this)">
              <option value="" disabled selected >Select a
                  Source Validation Dataset</option>
            {% for dd in dlist  %}
            <option value='{{dd}}' >{{dd}}</option>
            {% endfor %}
            </select>
            <label>Validation Dataset (Source)</label>
          </div>
        </div>

        <div class="row"> <!-- dd_tgt -->
          <div class="input-field col s6 light-blue-text text-
              accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
              ="Validation or Development Dataset for the
              Target Language">
            <select id = "dd_tgt" onchange="notify_language(
                this)">
              <option value="" disabled selected >Select a
                  Target Validation Dataset</option>
            {% for dd in dlist  %}
            <option value='{{dd}}' >{{dd}}</option>
            {% endfor %}
            </select>
            <label>Validation Dataset (Target)</label>
          </div>
        </div>

        <div class="row"> <!-- maxlen -->
          <div class="input-field col s6 light-blue-text text-
              accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
              ="Maximum Sentence Length for the Whole
              Training">
            <input id="maxlen" type="number" class="
                validate light-blue-text text-accent-4" min
                = "1" max = "200">
            <label for="maxlen" data-error="wrong" data-
                success="right">Maximum Sentence Length
                </label>
          </div>
        </div>

        <div class="row"> <!-- bpe_operations (for
            preprocessing) -->
          <div class="input-field col s6 light-blue-text text-
              accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
              ="Byte Pair Encoding Merge Operations">
            <input id="bpe_operations" type="number" class
                ="validate light-blue-text text-accent-4"
                min = "0">
            <label for="bpe_operations" data-error="wrong"
                data-success="right">BPE Merge Operations
                </label>
          </div>
        </div>

        <div class="row">
            <button href="{{url_for('preprocess')}}" class="
                btn waves-effect waves-light light-blue
                accent-4 btn-large right" name="preprocess
                ">Prepare Dataset
```

```
        </button>
    </div>
</div>
</li>
<li>
<div class="collapsible-header"><i class="material-icons
    ">location_city</i>Step 2: Select Architecture</div
    >
<div class="collapsible-body">
    <h5><span class = "light-blue-text text-accent-4">
        Select Architecture</span></h5>
    <div class="row">
        <div class="col s6">
          <div class="card medium sticky-action light-
              blue accent-4">
            <div class="card-image">
              <img id="arch_image" class="
                  materialboxed" src="{{url_for('.
                  static', filename='RNNSearch.
                  PNG')}}">
            </div>
            <div class="card-tabs">
              <ul class="tabs tabs-fixed-width">
                <li class="tab"><a href="#normal"
                    class="active"><span class="
                    light-blue-text text-accent
                    -4">Normal</span></a></li
                    >
                <li class="tab"><a class="" href
                    ="#advanced"><span class="
                    light-blue-text text-accent
                    -4">Advanced</span></a
                    ></li>
              <div class="indicator amber accent-2"
                  style="right: 239px; left: 0px; z-
                  index:1;"></div></ul>
            </div>
            <div class="card-content light-blue
                accent-4 white-text">
              <div id="normal">Use a Recurrent
                  Neural Network(RNN) with
                  Attention</div>
              <div id="advanced">Employ additional
                   mechanisms to the RNN with
              Attention to have more control of its
                   learning</div>
            </div>
            <a id="arch_image_normal" href="{{
                url_for('.static', filename='
                RNNSearch.PNG')}}" style="
                display: none;"></a>
            <a id="arch_image_advanced" href="{{
                url_for('.static', filename='Dropout.
                JPG')}}" style="display: none
                ;"></a>
        </div>
      </div>
  </div>
</div>
</li>
<li>
<div class="collapsible-header"><i class="material-icons
    ">build</i>Step 3: Generate Configuration</div>
<div class="collapsible-body">
    <h5><span class = "light-blue-text text-accent-4">
        Select Hyperparameters</span></h5>
    <div class="row"> <!-- suggest_defaults -->
        <div class="input-field col s3 light-blue-text text
            -accent-4">
          <label>Suggest Defaults?</label>
          <br>
        </div>
    </div>
    <div class="row"> <!-- suggest_defaults -->
      <div class="col s3 light-blue-text text-accent-4">
        <div class="switch tooltipped" data-position="
            right" data-delay="50" data-tooltip="
            This will add the default configuration of
            Nematus">
          <label>
            No
            <input type="checkbox" id="suggest_defaults
                " onchange="add_defaults(this)">
            <span class="lever"></span>
            Yes
          </label>
        </div>
      </div>
    </div>
    <div class="row"> <!-- optimizer -->
      <div class="input-field col s6 light-blue-text text-
          accent-4 tooltipped"
          data-position="right" data-delay="30" data-tooltip
          ="An optimizer to be used for the training">
        <select id = "optimizer">
          <option value="" disabled selected >Select an
              Optimizer</option>
          <option value="adam" >Adam</option>
          <option value="adadelta" >Adadelta</option>
          <option value="rmsprop" >RMS prop</option>
          <option value="sgd" >Stochastic Gradient
              Descent</option>
        </select>
        <label>Optimizer</label>
      </div>
    </div>

    <div class="row"> <!-- dim_word -->
      <div class="input-field col s6 light-blue-text text-
          accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
          ="The size of the layer that will represent the
          dictionary">
        <input id="dim_word" type="number" class="
            validate light-blue-text text-accent-4" min
            ="1">
        <label for="dim_word" data-error="wrong" data-
            success="right">Embedding Layer Size</
            label>
      </div>
    </div>

    <div class="row"> <!-- dim -->
      <div class="input-field col s6 light-blue-text text-
          accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
          ="The size of the layer that will learn the
          translation">
        <input id="dim" type="number" class="validate
            light-blue-text text-accent-4" min="1">
        <label for="dim" data-error="wrong" data-
            success="right">Hidden Layer Size</label>
      </div>
    </div>

    <div class="row"> <!-- lrate -->
      <div class="input-field col s6 light-blue-text text-
          accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
          ="The rate of learning of the model">
        <input id="lrate" type="number" class="validate
            light-blue-text text-accent-4" min = "0"
            max = "1" step="0.00001">
        <label for="lrate" data-error="wrong" data-
            success="right">Learning Rate</label>
      </div>
    </div>

    <div class="row"> <!-- n_words_src -->
      <div class="input-field col s6 light-blue-text text-
          accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
          ="Size of the source language vocabulary">
        <input id="n_words_src" type="number" class="
            validate light-blue-text text-accent-4" min
            ="10">
        <label for="n_words_src" data-error="wrong" data
            -success="right">Source Vocabulary Size</
            label>
      </div>
    </div>

    <div class="row"> <!-- n_words -->
      <div class="input-field col s6 light-blue-text text-
          accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
          ="Size of the target language vocabulary">
        <input id="n_words" type="number" class="
            validate light-blue-text text-accent-4" min
            ="10">
        <label for="n_words" data-error="wrong" data-
            success="right">Target Vocabulary Size</
            label>
      </div>
    </div>

    <div class="row"> <!-- batch_size -->
      <div class="input-field col s6 light-blue-text text-
          accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
          ="Number of sentences evaluated per batch">
        <input id="batch_size" type="number" class="
            validate light-blue-text text-accent-4" min
            = "1">
```

```
    <label for="batch_size" data-error="wrong" data-
        success="right">Minibatch Size</label>
  </div>
</div>

<div class="row"> <!-- valid_batch_size -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="Number of sentences evaluated per validation
      batch">
    <input id="valid_batch_size" type="number" class
        ="validate light-blue-text text-accent-4"
        min = "1">
    <label for="valid_batch_size" data-error="wrong"
        data-success="right">Validation Minibatch
        Size</label>
  </div>
</div>

<div class="row"> <!-- dispFreq -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="The frequency of display of loss or cost">
    <select id = "dispFreq">
      <option value="" disabled selected >Select an
          Interval</option>
      <option value="e" >Per Epoch</option>
      <option value="e1_2" >Per Half Epoch</option>
      <option value="e1_4" >Per Quarter Epoch</
          option>
      <option value="e1_8" >Per Eight Epoch</option
          >
    </select>
    <label>Loss Display Frequency</label>
  </div>
</div>

<div class="row"> <!-- validFreq -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="The frequency of validation. Best BLEU
      model is taken from here.">
    <select id = "validFreq">
      <option value="" disabled selected >Select an
          Interval</option>
      <option value="e5" >Per 5 Epochs</option>
      <option value="e5_2" >Per 2.5 Epochs</option>
      <option value="e" >Per Epoch</option>
      <option value="e1_2" >Per Half Epoch</option>
    </select>
    <label>Validation Frequency</label>
  </div>
</div>

<div class="row"> <!-- saveFreq -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="The frequency of saving of the model">
    <select id = "saveFreq">
      <option value="" disabled selected >Select an
          Interval</option>
      <option value="e10" >Per 10 Epochs</option>
      <option value="e5" >Per 5 Epochs</option>
      <option value="e5_2" >Per 2.5 Epochs</option>
      <option value="e" >Per Epoch</option>
    </select>
    <label>Save Frequency</label>
  </div>
</div>

<div class="row"> <!-- sampleFreq -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="The frequency where Nematus will display
      stochastically constructed sentences">
    <select id = "sampleFreq">
      <option value="" disabled selected >Select an
          Interval</option>
      <option value="e10" >Per 10 Epochs</option>
      <option value="e5" >Per 5 Epochs</option>
      <option value="e5_2" >Per 2.5 Epochs</option>
      <option value="e" >Per Epoch</option>
    </select>
    <label>Sample Frequency</label>
  </div>
</div>

<div class="row"> <!-- finish_after -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="Max number of batches until the training
      stops">
    <input id="finish_after" type="number" class="
        validate light-blue-text text-accent-4" min
        = "1">
    <label for="finish_after" data-error="wrong" data
        -success="right">Max Minibatches</label>
  </div>
</div>

<div class="row"> <!-- max_epochs -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="Max number of epochs until the training
      stops">
    <input id="max_epochs" type="number" class="
        validate light-blue-text text-accent-4" min
        = "1">
    <label for="max_epochs" data-error="wrong" data
        -success="right">Max Epochs</label>
  </div>
</div>

<h5><span class = "light-blue-text text-accent-4">
    Additional Mechanisms</span></h5>
<div class="row"> <!-- decay_c -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="Penalty to weights">
    <input id="decay_c" type="number" class="
        validate light-blue-text text-accent-4" min
        = "0.0" step="0.000001" disabled>
    <label for="decay_c" data-error="wrong" data-
        success="right">L2 Regularization Penalty</
        label>
  </div>
</div>

<div class="row"> <!-- map_decay_c -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="Penalty to original weights">
    <input id="map_decay_c" type="number" class="
        validate light-blue-text text-accent-4" min
        = "0.0" step="0.000001" disabled>
    <label for="map_decay_c" data-error="wrong"
        data-success="right">L2 Regularization
        Penalty (Original Weights)</label>
  </div>
</div>

<div class="row"> <!-- alpha_c -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="Adjustment to the alignment values">
    <input id="alpha_c" type="number" class="
        validate light-blue-text text-accent-4" min
        = "0.0" step="0.000001" disabled>
    <label for="alpha_c" data-error="wrong" data-
        success="right">Alignment Regularization</
        label>
  </div>
</div>

<div class="row"> <!-- clip_c -->
  <div class="input-field col s6 light-blue-text text-
      accent-4 tooltipped"
  data-position="right" data-delay="30" data-tooltip
      ="A technique to prevent exploding gradients">
    <input id="clip_c" type="number" class="validate
        light-blue-text text-accent-4" min="0.0"
        step="0.000001" disabled>
    <label for="clip_c" data-error="wrong" data-
        success="right">Gradient Clipping Threshold
        </label>
  </div>
</div>

<div class="row"> <!-- use_dropout -->
    <div class="input-field col s3 light-blue-text text
        -accent-4">
      <label>Use Dropout</label>
    </div>
</div>
```

```
    <div class="row"> <!-- use_dropout -->
      <div class="input-field col s6 light-blue-text text-
            accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
            ="Enables Dropout: A technique to prevent
            overfitting">
          <div class="switch">
            <label>
              False
              <input type="checkbox" id="use_dropout"
                    onchange="toggle_dropout(this)"
                    disabled>
              <span class="lever"></span>
              True
            </label>
          </div>
      </div>
    </div>
    <br>

    <div class="row"> <!-- dropout_embedding -->
      <div class="input-field col s6 light-blue-text text-
            accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
            ="Applies dropout on the dictionary
            representation">
        <input id="dropout_embedding" type="number"
              class="validate light-blue-text text-accent
              -4" min="0" step="0.001" disabled>
        <label for="dropout_embedding" data-error="
              wrong" data-success="right">Dropout for
              Input Embeddings</label>
      </div>
    </div>

    <div class="row"> <!-- dropout_hidden -->
      <div class="input-field col s6 light-blue-text text-
            accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
            ="Applies dropout on the layer that learns the
            translation">
        <input id="dropout_hidden" type="number" class
              ="validate light-blue-text text-accent-4"
              min="0" step="0.001" disabled>
        <label for="dropout_hidden" data-error="wrong"
              data-success="right">Dropout for Hidden
              Layer</label>
      </div>
    </div>

    <div class="row"> <!-- dropout_source -->
      <div class="input-field col s6 light-blue-text text-
            accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
            ="Applies dropout on the encoder">
        <input id="dropout_source" type="number" class
              ="validate light-blue-text text-accent-4"
              min="0" step="0.001" disabled>
        <label for="dropout_source" data-error="wrong"
              data-success="right">Dropout for Encoder (
              Source Words)</label>
      </div>
    </div>

    <div class="row"> <!-- dropout_target -->
      <div class="input-field col s6 light-blue-text text-
            accent-4 tooltipped"
      data-position="right" data-delay="30" data-tooltip
            ="Applies dropout on the decoder">
        <input id="dropout_target" type="number" class
              ="validate light-blue-text text-accent-4"
              min="0" step="0.001" disabled>
        <label for="dropout_target" data-error="wrong"
              data-success="right">Dropout for Decoder (
              Target Words)</label>
      </div>
    </div>
    <div class="row">
        <button href="{{url_for('generateConfig')}}" class
              ="btn waves-effect waves-light light-blue
              accent-4 btn-large right" name="
              generateConfig">Generate Configuration
        </button>
    </div>

</div>
</li>
<li>
<div class="collapsible-header"><i class="material-icons
      ">directions_run</i>Step 4: Start Training</div>
<div class="collapsible-body">
    <div class="row">
        <p>
        The training will start upon the click of the button
              below. Loss and BLEU scores will be displayed in
              the homepage.
        </p>
        </div>
        <div class="row">
        <button href="{{url_for('startTraining')}}" class="btn
              waves-effect waves-light light-blue accent-4 btn
              -large right" name="startTraining">Start
              Training
        </button>
        </div>
    </div>
    </li>
    </ul>
</div>
{% endblock %}


{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('.static', filename='
      style.css')}}">
{% endblock %}

{% block scripts %}
{{ super() }}
<script src="{{url_for('.static', filename='scripts_train.js')
      }}"></script>
{% endblock %}

File: neuro-mt\templates\translate.html

{# This simple template derives from ``base.html``. See ``base.
      html`` for
   more information about template inheritance. #}
{% extends "bootstrap_base.html" %}

{# Loads some of the macros included with Flask-Bootstrap.
      We are using the
   utils module here to automatically render Flask's flashed
         messages in a
   bootstrap friendly manner #}
{% import "bootstrap/utils.html" as utils %}


{# Inside the ``content`` is where you should place most of
      your own stuff.
   This will keep scripts at the page end and a navbar you add
         on later
   intact. #}
{% block content %}
<form action = "http://localhost:5000/translate" method = "
      POST"
        enctype = "multipart/form-data">
<div class="container-fluid text-center">
  <div class="row">
        <div class="col-sm-4">
        </div>
        <div class="col-sm-1">
    <input type="hidden" id="language-one-value" value="fil
          ">
        <span class="text btn-align" id="language-one">
              Filipino</span>
        </div>
        <div class="col-sm-2">
              <div class="center-block">
              <button type="button" class="btn btn-default
                    " style="background: none; border: none"
                    onclick="switch_language()" id="switcher
                    ">
              <img class="img-rounded" src="{{url_for('
                    static', filename='Arrow.png')}}" width
                    ="33" height="17"/>
            </button>
        </div>
        </div>
        <div class="col-sm-1">
    <input type="hidden" id="language-two-value" value="
          eng">
        <span class="text btn-align" id="language-two">
              English</span>
        </div>

        <div class="col-sm-2">
        <button type="button" class="btn btn-default" id="
              translate_button"
        style="
        border-width:1px;
        border-color:#FFD700;
        background-color:#3F51B5;
```

```
        color:#FFFFFF;
        " onclick="request_translation()">Translate</button>
        </div>
        <div class="col-sm-2">
        </div>
  </div>
  <br>
  <div class="row">
    <div class="col-sm-1">
    </div>
    <div class="col-sm-5 text-left">
      <textarea class="form-control" rows="8" id="input"
            style ="border-color: #808080; border-radius: 0;
            resize:none;" placeholder="Enter text"></textarea
            >
      <button type="button" class="btn btn-default pull-left
            " id="clear_button" style="border-color: #808080;
            border-radius: 0; border-top: 0;"
      onclick="clear_textarea()">Clear</button>
    </div>

    <div class="col-sm-5 text-left">
      <textarea class="form-control" rows="8" id="
            translation" readonly style="background: white;
            border-color: #FFD700; border-radius: 0; resize:
            none;"></textarea>
      <button type="button" class="btn btn-default" id="
            suggest_button" style="border-color: #FFD700;
            border-radius: 0; border-top: 0;"
      onclick="enable_suggest()">Suggest?</button>
      <button type="button" class="btn btn-default pull-left
            " id="suggest_cancel" style="border-color: #
            FFD700; border-radius: 0; border-top: 0; display:
            none;"
      onclick="disable_suggest()">Cancel</button>
      <button type="button" class="btn btn-default pull-
            right" id="suggest_submit" style="border-color: #
            FFD700; border-radius: 0; border-top: 0; display:
            none;"
      onclick="send_suggestion()">Submit</button>
    </div>
    <div class="col-sm-1">
    </div>
    </div>
  </div>
</div>
</form>
{% endblock %}
```

File: neuro-mt\templates\utils.html

```
{% macro flashed_messages(messages=None, container=True,
      transform={
  'critical': 'danger',
  'error': 'danger',
  'info': 'info',
  'warning': 'warning',
  'debug': 'default',
  'notset': 'default',
}, default_category=None) -%}
{% with messages = messages or get_flashed_messages(
      with_categories=True) -%}
{% if messages -%} {# don't output anything if there are no
      messages #}

{% if container -%}
<!-- begin message block -->
<div class="container">
  <div class="row">
    <div class="col-md-12">
{% endif -%}

{% for cat, msg in messages %}      <div class="alert alert-{{
      transform.get(cat.lower(), default_category or cat)}}" role
      ="alert">{{msg|safe}}</div>{% endfor -%}

{% if container %}
    </div>
  </div>
</div>
<!-- end message block -->
{% endif -%}

{% endif -%}
{% endwith -%}
{% endmacro -%}


{% macro icon(type=None, extra_classes=[]) -%}
<i{{ ({'class': (['mdi-' + type] + extra_classes)|join(' ')})|
      xmlattr }}{{ kwargs|xmlattr}}></i>
{%- endmacro %}
```

```
{% macro form_button(content, class = [], type='submit', name
      ='action', icon = False, iconclass=[]) -%}
<button class="btn {{ class|join(' ') }}" type="{{type}}"
      name="{{name}}">{{content}} {% if icon %}<i class
      ="{{ iconclass|join(' ') }} right"></i>{% endif %}</
      button>
{%- endmacro %}

{% macro card(title, content, actions = []) %}
<div class="card">
  <div class="card-content">
    <span class="card-title black-text">{{ title }}</span>
    {{ content|safe }}
  </div>
  {% if actions %}
  <div class="card-action">
  {% if actions %}
    {% for each in actions %}
      <a href="{{ each[0] }}">{{ each[1]|safe }}</a>
    {% endfor %}
  {% endif %}
  </div>
  {% endif %}
</div>
{% endmacro %}
```

File: neuro-mt\translation-scripts\oneTranslate-EngFil.sh

```
#!/bin/sh

# theano device, in case you do not want to compute on gpu,
      change it to cpu
device=gpu

# path to nematus ( https://www.github.com/rsennrich/
      nematus )
nematus=$4 #/path/to/nematus

SRC=eng

echo $1 > pendingTranslation.$2.$SRC

start=`date +%s`

./translation-scripts/preprocessOneTranslateEngFil.sh
      pendingTranslation.$2 $3

touch translation.output.$2.$SRC

THEANO_FLAGS=mode=FAST_RUN,floatX=float32,lib.cnmem
      =0.2,device=$device,on_unused_input=warn python
      $nematus/nematus/translate.py \
      -m $3 \
      -i pendingTranslation.$2.bpe.$SRC \
      -o translation.output.$2.$SRC \
      -k 12 -n -p 1

#k = 12 Beam size
#p = no. of processes 1 talaga

end=`date +%s`
./translation-scripts/postprocessOneTranslateEngFil.sh <
      translation.output.$2.$SRC > translation.output.
      preprocessed.$2.$SRC

runtime=$((end-start))
#echo Used the model $prefix
#printf "Your Input: "
#echo $1
#printf "Our Translations: "
while read p; do
  echo $p
done <translation.output.preprocessed.$2.$SRC
#echo Translated in $runtime seconds

rm pendingTranslation.$2.$SRC
rm pendingTranslation.$2.tok.$SRC
rm pendingTranslation.$2.tc.$SRC
rm pendingTranslation.$2.bpe.$SRC
rm translation.output.$2.$SRC
rm translation.output.preprocessed.$2.$SRC
```

File: neuro-mt\translation-scripts\oneTranslate-FilEng.sh

```
#!/bin/sh

# theano device, in case you do not want to compute on gpu,
      change it to cpu
device=gpu
```

```
# path to nematus ( https://www.github.com/rsennrich/
      nematus )
nematus=$4 #/path/to/nematus

SRC=fil

echo $1 > pendingTranslation.$2.$SRC

start=`date +%s`

./translation-scripts/preprocessOneTranslateFilEng.sh
      pendingTranslation.$2 $3

touch translation.output.$2.$SRC

THEANO_FLAGS=mode=FAST_RUN,floatX=float32,lib.cnmem
      =0.2,device=$device,on_unused_input=warn python
      $nematus/nematus/translate.py \
      -m $3 \
      -i pendingTranslation.$2.bpe.$SRC \
      -o translation.output.$2.$SRC \
      -k 12 -n -p 1

#k = 12 Beam size
#p = no. of processes 1 talaga

end=`date +%s`
./translation-scripts/postprocessOneTranslateFilEng.sh <
      translation.output.$2.$SRC > translation.output.
      preprocessed.$2.$SRC

runtime=$((end-start))
#echo Used the model model.npz.npz.best_bleu
#printf "Your Input: "
#echo $1
#printf "Our Translations: "
while read p; do
  echo $p
done <translation.output.preprocessed.$2.$SRC
#echo Translated in $runtime seconds

rm pendingTranslation.$2.$SRC
rm pendingTranslation.$2.tok.$SRC
rm pendingTranslation.$2.tc.$SRC
rm pendingTranslation.$2.bpe.$SRC
rm translation.output.$2.$SRC
rm translation.output.preprocessed.$2.$SRC
```

File: neuro-mt\translation-scripts\postprocessOneTranslateEngFil.sh

```
#/bin/sh

# path to moses decoder: https://github.com/moses-smt/
      mosesdecoder
moses_scripts=moses-scripts

# suffix of target language files
lng=fil

sed 's/\@\@ //g' | \
perl $moses_scripts/recaser/detruecase.perl | \
$moses_scripts/tokenizer/detokenizer.perl -l $lng
```

File: neuro-mt\translation-scripts\postprocessOneTranslateFilEng.sh

```
#/bin/sh

# path to moses decoder: https://github.com/moses-smt/
      mosesdecoder
moses_scripts=moses-scripts

# suffix of target language files
lng=eng

sed 's/\@\@ //g' | \
perl $moses_scripts/recaser/detruecase.perl | \
$moses_scripts/tokenizer/detokenizer.perl -l $lng
```

File: neuro-mt\translation-scripts\preprocessOneTranslateEngFil.sh

```
#!/bin/sh

# this sample script preprocesses a sample corpus, including
      tokenization,
# truecasing, and subword segmentation.
# for application to a different language pair,
# change source and target prefix, optionally the number of
      BPE operations,
# and the file names (currently, data/corpus and data/
      newsdev2016 are being processed)

# in the tokenization step, you will want to remove Romanian
      -specific normalization / diacritic removal,
# and you may want to add your own.
# also, you may want to learn BPE segmentations separately for
      each language,
# especially if they differ in their alphabet

# suffix of source language files
SRC=eng

# suffix of target language files
TRG=fil

# path to moses decoder: https://github.com/moses-smt/
      mosesdecoder
moses_scripts=moses-scripts

# path to subword segmentation scripts: https://github.com/
      rsennrich/subword-nmt
subword_nmt=subword-nmt-master

# path to nematus ( https://www.github.com/rsennrich/
      nematus )

# tokenize
for prefix in $1
 do
   cat $prefix.$SRC | \
   perl $moses_scripts/tokenizer/normalize-punctuation.perl -l
        $SRC | \
   perl $moses_scripts/tokenizer/tokenizer.perl -a -l $SRC >
        $prefix.tok.$SRC

 done

# apply truecaser (dev/test files)
for prefix in $1
 do
  perl $moses_scripts/recaser/truecase.perl -model truecasers/
       truecase-model.$SRC < $prefix.tok.$SRC > $prefix.tc.
       $SRC
 done

# apply BPE

for prefix in $1
 do
  python $subword_nmt/apply_bpe.py -c $2.bpe < $prefix.tc.
       $SRC > $prefix.bpe.$SRC
 done
```

File: neuro-mt\translation-scripts\preprocessOneTranslateFilEng.sh

```
#!/bin/sh

# this sample script preprocesses a sample corpus, including
      tokenization,
# truecasing, and subword segmentation.
# for application to a different language pair,
# change source and target prefix, optionally the number of
      BPE operations,
# and the file names (currently, data/corpus and data/
      newsdev2016 are being processed)

# in the tokenization step, you will want to remove Romanian
      -specific normalization / diacritic removal,
# and you may want to add your own.
# also, you may want to learn BPE segmentations separately for
      each language,
# especially if they differ in their alphabet

# suffix of source language files
SRC=fil

# suffix of target language files
TRG=eng

# path to moses decoder: https://github.com/moses-smt/
      mosesdecoder
moses_scripts=moses-scripts

# path to subword segmentation scripts: https://github.com/
      rsennrich/subword-nmt
subword_nmt=subword-nmt-master

# path to nematus ( https://www.github.com/rsennrich/
```

```
        nematus )

# tokenize
for prefix in $1
 do
    cat $prefix.$SRC | \
    perl $moses_scripts/tokenizer/normalize-punctuation.perl -l
          $SRC | \
    perl $moses_scripts/tokenizer/tokenizer.perl -a -l $SRC >
          $prefix.tok.$SRC

 done

# apply truecaser (dev/test files)
for prefix in $1
 do
  perl $moses_scripts/recaser/truecase.perl -model truecasers/
        truecase-model.$SRC < $prefix.tok.$SRC > $prefix.tc.
        $SRC
 done

# apply BPE
for prefix in $1
 do
  python $subword_nmt/apply_bpe.py -c $2.bpe < $prefix.tc.
        $SRC > $prefix.bpe.$SRC
 done
```

## A..2 Mobile App Codes

File: NeuroTranslate\.gitignore

```
*.iml
.gradle
/local.properties
/.idea/workspace.xml
/.idea/libraries
.DS_Store
/build
/captures
.externalNativeBuild
```

File: NeuroTranslate\build.gradle

```
// Top-level build file where you can add configuration options
       common to all sub-projects/modules.

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.2.3'

        // NOTE: Do not place your application dependencies
              here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

File: NeuroTranslate\gradle.properties

```
# Project-wide Gradle settings.

# IDE (e.g. Android Studio) users:
# Gradle settings configured through the IDE *will override*
# any settings specified in this file.

# For more details on how to configure your build environment
       visit
# http://www.gradle.org/docs/current/userguide/
       build_environment.html

# Specifies the JVM arguments used for the daemon process.
# The setting is particularly useful for tweaking memory
       settings.
org.gradle.jvmargs=-Xmx1536m

# When configured, Gradle will run in incubating parallel mode.
# This option should only be used with decoupled projects.
       More details, visit
# http://www.gradle.org/docs/current/userguide/
       multi_project_builds.html#sec:decoupled_projects
# org.gradle.parallel=true
```

File: NeuroTranslate\gradlew

```
#!/usr/bin/env bash

##############################################################

##
## Gradle start up script for UN*X
##
##############################################################


# Add default JVM options here. You can also use JAVA_OPTS
       and GRADLE_OPTS to pass JVM options to this script.
DEFAULT_JVM_OPTS=""

APP_NAME="Gradle"
APP_BASE_NAME=`basename "$0"`

# Use the maximum available, or set MAX_FD != -1 to use
       that value.
MAX_FD="maximum"

warn () {
    echo "$*"
}

die () {
    echo
    echo "$*"
    echo
    exit 1
}

# OS specific support (must be 'true' or 'false').
cygwin=false
msys=false
darwin=false
case "`uname`" in
  CYGWIN* )
    cygwin=true
    ;;
  Darwin* )
    darwin=true
    ;;
  MINGW* )
    msys=true
    ;;
esac

# Attempt to set APP_HOME
# Resolve links: $0 may be a link
PRG="$0"
# Need this for relative symlinks.
while [ -h "$PRG" ] ; do
    ls=`ls -ld "$PRG"`
    link=`expr "$ls" : '.*-> \(.*\)$'`
    if expr "$link" : '/.*' > /dev/null; then
        PRG="$link"
    else
        PRG=`dirname "$PRG"`"/$link"
    fi
done
SAVED="`pwd`"
cd "`dirname \"$PRG\"`/" >/dev/null
APP_HOME="`pwd -P`"
cd "$SAVED" >/dev/null

CLASSPATH=$APP_HOME/gradle/wrapper/gradle-wrapper.
       jar

# Determine the Java command to use to start the JVM.
if [ -n "$JAVA_HOME" ] ; then
    if [ -x "$JAVA_HOME/jre/sh/java" ] ; then
        # IBM's JDK on AIX uses strange locations for the
              executables
        JAVACMD="$JAVA_HOME/jre/sh/java"
    else
        JAVACMD="$JAVA_HOME/bin/java"
    fi
    if [ ! -x "$JAVACMD" ] ; then
        die "ERROR: JAVA_HOME is set to an invalid directory
              : $JAVA_HOME

Please set the JAVA_HOME variable in your environment to
```

```
        match the
location of your Java installation."
    fi
else
    JAVACMD="java"
    which java >/dev/null 2>&1 || die "ERROR: JAVA_HOME
        is not set and no 'java' command could be found in
        your PATH.

Please set the JAVA_HOME variable in your environment to
        match the
location of your Java installation."
fi

# Increase the maximum file descriptors if we can.
if [ "$cygwin" = "false" -a "$darwin" = "false" ] ; then
    MAX_FD_LIMIT=`ulimit -H -n`
    if [ $? -eq 0 ] ; then
        if [ "$MAX_FD" = "maximum" -o "$MAX_FD" = "
            max" ] ; then
            MAX_FD="$MAX_FD_LIMIT"
        fi
        ulimit -n $MAX_FD
        if [ $? -ne 0 ] ; then
            warn "Could not set maximum file descriptor limit:
                $MAX_FD"
        fi
    else
        warn "Could not query maximum file descriptor limit:
            $MAX_FD_LIMIT"
    fi
fi

# For Darwin, add options to specify how the application
    appears in the dock
if $darwin; then
    GRADLE_OPTS="$GRADLE_OPTS \"-Xdock:name=
        $APP_NAME\" \"-Xdock:icon=$APP_HOME/media/
        gradle.icns\""
fi

# For Cygwin, switch paths to Windows format before running
    java
if $cygwin ; then
    APP_HOME=`cygpath --path --mixed "$APP_HOME"`
    CLASSPATH=`cygpath --path --mixed "$CLASSPATH"`
    JAVACMD=`cygpath --unix "$JAVACMD"`

    # We build the pattern for arguments to be converted via
        cygpath
    ROOTDIRSRAW=`find -L / -maxdepth 1 -mindepth 1 -
        type d 2>/dev/null`
    SEP=""
    for dir in $ROOTDIRSRAW ; do
        ROOTDIRS="$ROOTDIRS$SEP$dir"
        SEP="|"
    done
    OURCYGPATTERN="(^($ROOTDIRS))"
    # Add a user-defined pattern to the cygpath arguments
    if [ "$GRADLE_CYGPATTERN" != "" ] ; then
        OURCYGPATTERN="$OURCYGPATTERN|(
            $GRADLE_CYGPATTERN)"
    fi
    # Now convert the arguments - kludge to limit ourselves to
        /bin/sh
    i=0
    for arg in "$@" ; do
        CHECK=`echo "$arg"|egrep -c "$OURCYGPATTERN"
            -`
        CHECK2=`echo "$arg"|egrep -c "^-"`
                                    ### Determine if an
            option

        if [ $CHECK -ne 0 ] && [ $CHECK2 -eq 0 ] ; then
                                ### Added a condition
            eval `echo args$i=`cygpath --path --ignore --
                mixed "$arg"`
        else
            eval `echo args$i="\"$arg\""`
        fi
        i=$((i+1))
    done
    case $i in
        (0) set -- ;;
        (1) set -- "$args0" ;;
        (2) set -- "$args0" "$args1" ;;
        (3) set -- "$args0" "$args1" "$args2" ;;
        (4) set -- "$args0" "$args1" "$args2" "$args3" ;;
        (5) set -- "$args0" "$args1" "$args2" "$args3" "
            $args4" ;;
        (6) set -- "$args0" "$args1" "$args2" "$args3" "
            $args4" "$args5" ;;
        (7) set -- "$args0" "$args1" "$args2" "$args3" "
            $args4" "$args5" "$args6" ;;
        (8) set -- "$args0" "$args1" "$args2" "$args3" "
            $args4" "$args5" "$args6" "$args7" ;;
        (9) set -- "$args0" "$args1" "$args2" "$args3" "
            $args4" "$args5" "$args6" "$args7" "$args8" ;;
    esac
fi

# Split up the JVM_OPTS And GRADLE_OPTS values into an
    array, following the shell quoting and substitution rules
function splitJvmOpts() {
    JVM_OPTS=("$@")
}
eval splitJvmOpts $DEFAULT_JVM_OPTS $JAVA_OPTS
    $GRADLE_OPTS
JVM_OPTS[${#JVM_OPTS[*]}]="-Dorg.gradle.appname=
    $APP_BASE_NAME"

exec "$JAVACMD" "${JVM_OPTS[@]}" -classpath "
    $CLASSPATH" org.gradle.wrapper.GradleWrapperMain "
    $@"


File: NeuroTranslate\gradlew.bat

@if "%DEBUG%" == "" @echo off
@rem
    ##########################################################

@rem
@rem Gradle startup script for Windows
@rem
@rem
    ##########################################################


@rem Set local scope for the variables with windows NT shell
if "%OS%"=="Windows_NT" setlocal

@rem Add default JVM options here. You can also use
    JAVA_OPTS and GRADLE_OPTS to pass JVM options to
    this script.
set DEFAULT_JVM_OPTS=

set DIRNAME=%~dp0
if "%DIRNAME%" == "" set DIRNAME=.
set APP_BASE_NAME=%~n0
set APP_HOME=%DIRNAME%

@rem Find java.exe
if defined JAVA_HOME goto findJavaFromJavaHome

set JAVA_EXE=java.exe
%JAVA_EXE% -version >NUL 2>&1
if "%ERRORLEVEL%" == "0" goto init

echo.
echo ERROR: JAVA_HOME is not set and no 'java' command
    could be found in your PATH.
echo.
echo Please set the JAVA_HOME variable in your environment
    to match the
echo location of your Java installation.

goto fail

:findJavaFromJavaHome
set JAVA_HOME=%JAVA_HOME:"=%
set JAVA_EXE=%JAVA_HOME%/bin/java.exe

if exist "%JAVA_EXE%" goto init

echo.
echo ERROR: JAVA_HOME is set to an invalid directory: %
    JAVA_HOME%
echo.
echo Please set the JAVA_HOME variable in your environment
    to match the
echo location of your Java installation.

goto fail

:init
@rem Get command-line arguments, handling Windowz variants

if not "%OS%" == "Windows_NT" goto win9xME_args
if "%@eval[2+2]" == "4" goto 4NT_args

:win9xME_args
@rem Slurp the command line arguments.
set CMD_LINE_ARGS=
```

```
set _SKIP=2

:win9xME_args_slurp
if "x%~1" == "x" goto execute

set CMD_LINE_ARGS=%*
goto execute

:4NT_args
@rem Get arguments from the 4NT Shell from JP Software
set CMD_LINE_ARGS=%$

:execute
@rem Setup the command line

set CLASSPATH=%APP_HOME%\gradle\wrapper\gradle-
        wrapper.jar

@rem Execute Gradle
"%JAVA_EXE%" %DEFAULT_JVM_OPTS% %JAVA_OPTS% %
        GRADLE_OPTS% "-Dorg.gradle.appname=%
        APP_BASE_NAME%" -classpath "%CLASSPATH%" org.
        gradle.wrapper.GradleWrapperMain %CMD_LINE_ARGS%

:end
@rem End local scope for the variables with windows NT shell
if "%ERRORLEVEL%"=="0" goto mainEnd

:fail
rem Set variable GRADLE_EXIT_CONSOLE if you need the
        _script_ return code instead of
rem the _cmd.exe /c_ return code!
if  not "" == "%GRADLE_EXIT_CONSOLE%" exit 1
exit /b 1

:mainEnd
if "%OS%"=="Windows_NT" endlocal

:omega


File: NeuroTranslate\settings.gradle

include ':app'


File: NeuroTranslate\app\.gitignore

/build


File: NeuroTranslate\app\app.iml

<?xml version="1.0" encoding="UTF-8"?>
<module external.linked.project.id=":app" external.linked.
        project.path="$MODULE_DIR$" external.root.project.
        path="$MODULE_DIR$/.." external.system.id="
        GRADLE" type="JAVA_MODULE" version="4">
  <component name="FacetManager">
    <facet type="android-gradle" name="Android-Gradle">
      <configuration>
        <option name="GRADLE_PROJECT_PATH" value=":
            app" />
      </configuration>
    </facet>
    <facet type="android" name="Android">
      <configuration>
        <option name="SELECTED_BUILD_VARIANT" value
            ="debug" />
        <option name="SELECTED_TEST_ARTIFACT" value
            ="_android_test_" />
        <option name="ASSEMBLE_TASK_NAME" value="
            assembleDebug" />
        <option name="COMPILE_JAVA_TASK_NAME" value
            ="compileDebugSources" />
        <afterSyncTasks>
          <task>generateDebugSources</task>
        </afterSyncTasks>
        <option name="ALLOW_USER_CONFIGURATION"
            value="false" />
        <option name="MANIFEST_FILE_RELATIVE_PATH"
            value="/src/main/AndroidManifest.xml" />
        <option name="RES_FOLDER_RELATIVE_PATH"
            value="/src/main/res" />
        <option name="RES_FOLDERS_RELATIVE_PATH"
            value="file://$MODULE_DIR$/src/main/res" />
        <option name="ASSETS_FOLDER_RELATIVE_PATH"
            value="/src/main/assets" />
      </configuration>
    </facet>
  </component>
  <component name="NewModuleRootManager"
      LANGUAGE_LEVEL="JDK_1_7" inherit-compiler-
      output="false">
<output url="file://$MODULE_DIR$/build/intermediates/
      classes/debug" />
<output-test url="file://$MODULE_DIR$/build/
      intermediates/classes/test/debug" />
<exclude-output />
<content url="file://$MODULE_DIR$">
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/r/debug" isTestSource="false"
      generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/aidl/debug" isTestSource="false"
      generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/buildConfig/debug" isTestSource
      ="false" generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/rs/debug" isTestSource="false"
      generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/apt/debug" isTestSource="false"
      generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/res/rs/debug" type="java-resource" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/res/resValues/debug" type="java-
      resource" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/r/androidTest/debug"
      isTestSource="true" generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/aidl/androidTest/debug"
      isTestSource="true" generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/buildConfig/androidTest/debug"
      isTestSource="true" generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/rs/androidTest/debug"
      isTestSource="true" generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/source/apt/androidTest/debug"
      isTestSource="true" generated="true" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/res/rs/androidTest/debug" type="java-
      test-resource" />
  <sourceFolder url="file://$MODULE_DIR$/build/
      generated/res/resValues/androidTest/debug" type="
      java-test-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/res
      " type="java-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/
      resources" type="java-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/
      assets" type="java-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/
      aidl" isTestSource="false" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/
      java" isTestSource="false" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/jni
      " isTestSource="false" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/rs"
       isTestSource="false" />
  <sourceFolder url="file://$MODULE_DIR$/src/debug/
      shaders" isTestSource="false" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /res" type="java-test-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /resources" type="java-test-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /assets" type="java-test-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /aidl" isTestSource="true" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /java" isTestSource="true" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /jni" isTestSource="true" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /rs" isTestSource="true" />
  <sourceFolder url="file://$MODULE_DIR$/src/testDebug
      /shaders" isTestSource="true" />
  <sourceFolder url="file://$MODULE_DIR$/src/main/res"
       type="java-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/main/
      resources" type="java-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/main/
      assets" type="java-resource" />
  <sourceFolder url="file://$MODULE_DIR$/src/main/aidl
      " isTestSource="false" />
  <sourceFolder url="file://$MODULE_DIR$/src/main/java
      " isTestSource="false" />
  <sourceFolder url="file://$MODULE_DIR$/src/main/jni"
```

```
        isTestSource="false" />
<sourceFolder url="file://$MODULE_DIR$/src/main/rs"
      isTestSource="false" />
<sourceFolder url="file://$MODULE_DIR$/src/main/
      shaders" isTestSource="false" />
<sourceFolder url="file://$MODULE_DIR$/src/test/res"
      type="java-test-resource" />
<sourceFolder url="file://$MODULE_DIR$/src/test/
      resources" type="java-test-resource" />
<sourceFolder url="file://$MODULE_DIR$/src/test/
      assets" type="java-test-resource" />
<sourceFolder url="file://$MODULE_DIR$/src/test/aidl"
      isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/test/java"
       isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/test/jni"
      isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/test/rs"
      isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/test/
      shaders" isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/res" type="java-test-resource" />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/resources" type="java-test-resource"
      />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/assets" type="java-test-resource" />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/aidl" isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/java" isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/jni" isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/rs" isTestSource="true" />
<sourceFolder url="file://$MODULE_DIR$/src/
      androidTest/shaders" isTestSource="true" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/assets" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/blame" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/builds" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/classes" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/dependency-cache" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support.
      test.espresso/espresso-core/2.2.2/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support.
      test.espresso/espresso-idling-resource/2.2.2/jars"
      />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support.
      test/exposed-instrumentation-api-publish/0.5/jars
      " />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support.
      test/rules/0.5/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support.
      test/runner/0.5/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      animated-vector-drawable/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      appcompat-v7/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      design/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      recyclerview-v7/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      support-compat/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      support-core-ui/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      support-core-utils/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      support-fragment/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      support-media-compat/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      support-v4/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      support-vector-drawable/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/exploded-aar/com.android.support/
      transition/25.1.0/jars" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/incremental" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/incremental-classes" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/incremental-runtime-classes" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/incremental-safeguard" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/incremental-verifier" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/instant-run-resources" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/instant-run-support" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/jniLibs" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/manifests" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/pre-dexed" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/reload-dex" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/res" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/restart-dex" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/rs" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/shaders" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/symbols" />
<excludeFolder url="file://$MODULE_DIR$/build/
      intermediates/transforms" />
<excludeFolder url="file://$MODULE_DIR$/build/
      outputs" />
<excludeFolder url="file://$MODULE_DIR$/build/tmp"
      />
</content>
<orderEntry type="jdk" jdkName="Android API 25
      Platform" jdkType="Android SDK" />
<orderEntry type="sourceFolder" forTests="false" />
<orderEntry type="library" exported="" name="design
      -25.1.0" level="project" />
<orderEntry type="library" exported="" scope="TEST"
      name="runner-0.5" level="project" />
<orderEntry type="library" exported="" scope="TEST"
      name="espresso-idling-resource-2.2.2" level="
      project" />
<orderEntry type="library" exported="" scope="TEST"
      name="hamcrest-library-1.3" level="project" />
<orderEntry type="library" exported="" scope="TEST"
      name="hamcrest-integration-1.3" level="project"
      />
<orderEntry type="library" exported="" name="
      androidannotations-4.2.0" level="project" />
<orderEntry type="library" exported="" name="transition
      -25.1.0" level="project" />
<orderEntry type="library" exported="" name="support-
      fragment-25.1.0" level="project" />
<orderEntry type="library" exported="" name="
      jcodemodel-2.8.5" level="project" />
<orderEntry type="library" exported="" name="support-
      core-ui-25.1.0" level="project" />
<orderEntry type="library" exported="" name="otto
      -1.3.8" level="project" />
<orderEntry type="library" exported="" scope="TEST"
      name="jsr305-2.0.1" level="project" />
<orderEntry type="library" exported="" name="support-
      core-utils-25.1.0" level="project" />
<orderEntry type="library" exported="" name="
      androidannotations-api-4.2.0" level="project" />
<orderEntry type="library" exported="" scope="TEST"
      name="espresso-core-2.2.2" level="project" />
<orderEntry type="library" exported="" name="support-
      annotations-25.1.0" level="project" />
<orderEntry type="library" exported="" scope="TEST"
      name="exposed-instrumentation-api-publish-0.5"
      level="project" />
<orderEntry type="library" exported="" scope="TEST"
      name="rules-0.5" level="project" />
<orderEntry type="library" exported="" name="jackson-
      databind-2.8.6" level="project" />
```

```
    <orderEntry type="library" exported="" scope="TEST"
        name="javax.annotation-api-1.2" level="project"
        />
    <orderEntry type="library" exported="" scope="TEST"
        name="javax.inject-1" level="project" />
    <orderEntry type="library" exported="" name="jackson-
        annotations-2.8.6" level="project" />
    <orderEntry type="library" exported="" name="okio
        -1.11.0" level="project" />
    <orderEntry type="library" exported="" name="support-
        compat-25.1.0" level="project" />
    <orderEntry type="library" exported="" name="animated
        -vector-drawable-25.1.0" level="project" />
    <orderEntry type="library" exported="" name="support-
        v4-25.1.0" level="project" />
    <orderEntry type="library" exported="" scope="TEST"
        name="javawriter-2.1.1" level="project" />
    <orderEntry type="library" exported="" scope="TEST"
        name="hamcrest-core-1.3" level="project" />
    <orderEntry type="library" exported="" name="
        annotations-2.0.3" level="project" />
    <orderEntry type="library" exported="" name="okhttp
        -3.5.0" level="project" />
    <orderEntry type="library" exported="" name="support-
        media-compat-25.1.0" level="project" />
    <orderEntry type="library" exported="" scope="TEST"
        name="junit-4.12" level="project" />
    <orderEntry type="library" exported="" name="support-
        vector-drawable-25.1.0" level="project" />
    <orderEntry type="library" exported="" name="jackson-
        core-2.8.6" level="project" />
    <orderEntry type="library" exported="" name="
        appcompat-v7-25.1.0" level="project" />
    <orderEntry type="library" exported="" name="
        recyclerview-v7-25.1.0" level="project" />
    <orderEntry type="library" exported="" name="otto
        -4.2.0" level="project" />
    <orderEntry type="library" exported="" name="android-
        android-25" level="project" />
  </component>
</module>
```

File: NeuroTranslate\app\build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "23.0.3"
    defaultConfig {
        applicationId "agtacorda.dpsm.cas.upm.neurotranslate"
        minSdkVersion 16
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.
            AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-
                android.txt'), 'proguard-rules.pro'
        }
    }
    packagingOptions {
        exclude 'META-INF/DEPENDENCIES'
        exclude 'META-INF/NOTICE'
        exclude 'META-INF/LICENSE'
        exclude 'META-INF/LICENSE.txt'
        exclude 'META-INF/NOTICE.txt'
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:
        espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support
            -annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.0'
    compile 'com.android.support:design:25.1.0'
    testCompile 'junit:junit:4.12'
    compile 'org.androidannotations:androidannotations:4.2.0'
    compile 'org.androidannotations:androidannotations-api
        :4.2.0'
    compile 'com.squareup:otto:1.3.8'
    compile 'org.androidannotations:otto:4.2.0'
    compile 'com.fasterxml.jackson.core:jackson-core:2.8.6'
    compile 'com.fasterxml.jackson.core:jackson-annotations
        :2.8.6'
    compile 'com.fasterxml.jackson.core:jackson-databind:2.8.6'
    compile 'com.squareup.okhttp3:okhttp:3.5.0'
}
```

File: NeuroTranslate\app\proguard-rules.pro

```
# Add project specific ProGuard rules here.
# By default, the flags in this file are appended to flags
    specified
# in C:\Users\Alfred\AppData\Local\Android\Sdk/tools/
    proguard/proguard-android.txt
# You can edit the include path and order by changing the
    proguardFiles
# directive in build.gradle.
#
# For more details, see
#   http://developer.android.com/guide/developing/tools/
    proguard.html

# Add any project specific keep options here:

# If your project uses WebView with JS, uncomment the
    following
# and specify the fully qualified class name to the JavaScript
    interface
# class:
#-keepclassmembers class fqcn.of.javascript.interface.for.
    webview {
#   public *;
#}
```

File: NeuroTranslate\app\src\main\AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res
    /android"
    package="agtacorda.dpsm.cas.upm.neurotranslate">

    <uses-permission android:name="android.permission.
        INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/neuromt_logo"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.
            NoActionBar">
        <activity
            android:name=".activities.TranslatorActivity_"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.
                    MAIN" />

                <category android:name="android.intent.
                    category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".activities.SuggestActivity_"
            android:parentActivityName=".activities.
                TranslatorActivity_"
            android:screenOrientation="portrait">

            <!-- Parent activity meta-data to support 4.0 and
                lower -->
            <meta-data
                android:name="android.support.
                    PARENT_ACTIVITY"
                android:value=".activities.TranslatorActivity_"
                    />
        </activity>
        <activity
            android:name=".activities.InfoActivity_"
            android:parentActivityName=".activities.
                TranslatorActivity_"
            android:screenOrientation="portrait">

            <!-- Parent activity meta-data to support 4.0 and
                lower -->
            <meta-data
                android:name="android.support.
                    PARENT_ACTIVITY"
                android:value=".activities.TranslatorActivity_"
                    />
        </activity>
    </application>

</manifest>
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\activities\InfoActivity.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.activities;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import org.androidannotations.annotations.Click;
import org.androidannotations.annotations.EActivity;

import agtacorda.dpsm.cas.upm.neurotranslate.R;

@EActivity(R.layout.activity_info)
public class InfoActivity extends AppCompatActivity{
    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onStart(){
        super.onStart();
    }

    @Override
    public void onStop(){
        super.onStop();
    }

    @Override
    public void onDestroy(){
        super.onDestroy();
    }

    @Click(R.id.ib_back)
    void backClick(){
        onBackPressed();
    }
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\activities\SuggestActivity.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.activities;

import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import com.squareup.otto.Subscribe;

import org.androidannotations.annotations.AfterViews;
import org.androidannotations.annotations.Bean;
import org.androidannotations.annotations.Click;
import org.androidannotations.annotations.EActivity;
import org.androidannotations.annotations.EditorAction;
import org.androidannotations.annotations.Extra;
import org.androidannotations.annotations.TextChange;
import org.androidannotations.annotations.ViewById;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

import agtacorda.dpsm.cas.upm.neurotranslate.R;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.APIManager;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.Constants;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.EventsManager;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.OttoBus;

@EActivity(R.layout.activity_suggest)
public class SuggestActivity extends AppCompatActivity {

    @ViewById(R.id.ib_back)
    ImageButton backView;

    @ViewById(R.id.et_suggestion)
    EditText suggestionView;

    @ViewById(R.id.tv_input)
    TextView inputView;

    @Extra("language_mode")
    int language_mode;

    @Extra("translation")
    String translation;

    @Extra("input")
    String input;

    @Bean
    APIManager apiManager;

    @Bean
    OttoBus bus;

    @AfterViews
    void initialize () {
        suggestionView.setText(translation);
        suggestionView.setSelection(suggestionView.getText().length());
        inputView.setText(input);
    }

    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onStart(){
        super.onStart();
        bus.register(this);
    }

    @Override
    public void onStop(){
        super.onStop();
        bus.unregister(this);
    }

    @Override
    public void onDestroy(){
        super.onDestroy();
    }

    //enter
    @EditorAction(R.id.et_suggestion) //optimize http://stackoverflow.com/questions/27845069/add-a-new-item-to-recyclerview-programatically
    void onEditTextAction(TextView input, int actionId, KeyEvent keyEvent) {
        if (actionId == EditorInfo.IME_NULL
                || keyEvent == null
                || keyEvent.getKeyCode() == KeyEvent.KEYCODE_ENTER) {
            suggestTranslation(input.getText().toString());
            InputMethodManager in = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
            in.hideSoftInputFromWindow(input.getApplicationWindowToken(), InputMethodManager.HIDE_NOT_ALWAYS);
        }
    }

    //newline
    @TextChange(R.id.et_suggestion)
    void onTextChange(CharSequence text, TextView input, int before, int start, int count) {
        String string = text.toString();
        if (string.length() > 0 && string.charAt(string.length() - 1) == '\n') {
            suggestTranslation(input.getText().toString().replace("\n", ""));
            InputMethodManager in = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
            in.hideSoftInputFromWindow(input.getApplicationWindowToken(), InputMethodManager.HIDE_NOT_ALWAYS);
        }
    }

    private void suggestTranslation(String suggestion){
        Map<String, String> dataMap = new HashMap<String,
```

```
                String>();
        if(language_mode == 0){
            dataMap.put("fil", input);
            dataMap.put("eng", suggestion);
        }
        else  if(language_mode == 1){
            dataMap.put("eng", input);
            dataMap.put("fil", suggestion);
        }
        JSONObject data = new JSONObject(dataMap);
        Log.i("INFO", data.toString());

        apiManager.suggestTranslation(data);
    }

    @Subscribe
    public void suggestTranslationReturn(EventsManager.
          SuggestTranslationEvent e){
        if(e.status == Constants.STATUS_CODE_SUCCESS){
            Toast.makeText(this, "Thank You for the suggestion
                  !", Toast.LENGTH_SHORT).show();
            goBack();
        }
        else{
            Toast.makeText(this, "Failed to send suggestion.",
                  Toast.LENGTH_SHORT).show();
        }
    }

    @Click(R.id.ib_back)
    void goBack() {
        onBackPressed();
    }
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\activities\TranslatorActivity.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.activities;

import android.content.Context;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.animation.RotateAnimation;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import com.squareup.otto.Subscribe;

import org.androidannotations.annotations.AfterViews;
import org.androidannotations.annotations.Bean;
import org.androidannotations.annotations.Click;
import org.androidannotations.annotations.EActivity;
import org.androidannotations.annotations.EditorAction;
import org.androidannotations.annotations.TextChange;
import org.androidannotations.annotations.ViewById;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import agtacorda.dpsm.cas.upm.neurotranslate.R;
import agtacorda.dpsm.cas.upm.neurotranslate.adapters.
       TranslationsAdapter;
import agtacorda.dpsm.cas.upm.neurotranslate.extra.
       ItemClickSupport;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.
       APIManager;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.
       Constants;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.
       EventsManager;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.
       OttoBus;
import agtacorda.dpsm.cas.upm.neurotranslate.views.
       SettingsView;

/**
 * Created by Alfred on 1/20/2017.
 */
@EActivity(R.layout.activity_translator)
public class TranslatorActivity extends AppCompatActivity {

    @ViewById(R.id.ib_info)
    ImageButton infoView;

    @ViewById(R.id.ib_switch)
    ImageButton switchView;

    @ViewById(R.id.tv_language_1)
    TextView languageOneView;

    @ViewById(R.id.tv_language_2)
    TextView languageTwoView;

    @ViewById(R.id.et_input)
    EditText inputView;

    @ViewById(R.id.rv_translation_list)
    RecyclerView translationListView;

    @ViewById(R.id.settings_view)
    SettingsView settingsView;

    @Bean
    APIManager apiManager;

    @Bean
    OttoBus bus;

    ArrayList<String> translations = new ArrayList<String>();
          //list of requested translations fil-eng
    ArrayList<String> locations = new ArrayList<String>();

    ArrayList<String> translations2 = new ArrayList<String
         >(); //list of requested translations eng-fil
    ArrayList<String> locations2 = new ArrayList<String>();

    TranslationsAdapter translationsAdapter;

    public static final String preferences = "
          neurotranslatePreferences" ;

    private int mCurrRotation = 0;

    private static int HISTORY_COUNT = 10;

    private int language_mode = 0;

    private String current_click_input = "";
    public static SharedPreferences prefs;

    @AfterViews
    void initialize () {
        translationListView.setHasFixedSize(true);
        LinearLayoutManager layoutManager = new
              LinearLayoutManager(this);
        layoutManager.setOrientation(LinearLayoutManager.
              VERTICAL);
        layoutManager.setReverseLayout(true);
        layoutManager.setStackFromEnd(true);
        translationListView.setLayoutManager(layoutManager);
        settingsView.setVisibility(View.GONE);
    }

    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        prefs = TranslatorActivity.this.getSharedPreferences(
              preferences, MODE_PRIVATE);
        if(prefs.contains("max")){
            HISTORY_COUNT = Integer.parseInt(prefs.
                  getString("max","10"));
        }
        if(prefs.contains("address")){
            Constants.API_ADDRESS = prefs.getString("
                  address", Constants.API_ADDRESS);
            Constants.API_ROOT = "http://" + Constants.
                  API_ADDRESS;
            Constants.API_TRANSLATE = Constants.
                  API_ROOT + "/translate";
            Constants.API_SUGGEST = Constants.API_ROOT
                  + "/suggest";
        }
    }

    @Override
    public void onStart(){
        super.onStart();
        bus.register(this);
```

```
}

@Override
public void onStop(){
    super.onStop();
    bus.unregister(this);
}

@Override
public void onDestroy(){
    super.onDestroy();
}

//enter
@EditorAction(R.id.et_input) //optimize http://
      stackoverflow.com/questions/27845069/add-a-new-
      item-to-recyclerview-programatically
void onEditTextAction(TextView input, int actionId,
      KeyEvent keyEvent) {
    if (actionId == EditorInfo.IME_NULL
        || keyEvent == null
        || keyEvent.getKeyCode() == KeyEvent.
            KEYCODE_ENTER) {
        requestTranslation(input.getText().toString());
        input.setText("");
        input.clearFocus();
        InputMethodManager in = (InputMethodManager)
            getSystemService(Context.
            INPUT_METHOD_SERVICE);
        in.hideSoftInputFromWindow(input
                .getApplicationWindowToken(),
            InputMethodManager.HIDE_NOT_ALWAYS)
                ;
    }
}

//newline
@TextChange(R.id.et_input)
void onTextChange(CharSequence text, TextView input, int
      before, int start, int count) {
    String string = text.toString();
    if (string.length() > 0 && string.charAt(string.length
        () - 1) == '\n') {
        requestTranslation(input.getText().toString().
            replace("\n", ""));
        input.setText("");
        input.clearFocus();
        InputMethodManager in = (InputMethodManager)
            getSystemService(Context.
            INPUT_METHOD_SERVICE);
        in.hideSoftInputFromWindow(input
                .getApplicationWindowToken(),
            InputMethodManager.HIDE_NOT_ALWAYS)
                ;
    }
}

private void requestTranslation(String input){
    Map<String, String> dataMap = new HashMap<String,
        String>();
    if(language_mode == 0){
        dataMap.put("source", "fil");
        dataMap.put("target", "eng");
    }
    else if(language_mode == 1){
        dataMap.put("source", "eng");
        dataMap.put("target", "fil");
    }
    dataMap.put("input", input);
    JSONObject data = new JSONObject(dataMap);
    Log.i("INFO", data.toString());

    apiManager.requestTranslation(data,input);
}

@Subscribe
public void updatePrefs(EventsManager.PreferencesUpdated
      e){
    if(prefs.contains("max")){
        if(Integer.parseInt(prefs.getString("max","10")) >
            0) {
            HISTORY_COUNT = Integer.parseInt(prefs.
                getString("max", "10"));
        }
    }
    if(prefs.contains("address")){
        Constants.API_ADDRESS = prefs.getString("
            address", Constants.API_ADDRESS);
        Constants.API_ROOT = "http://" + Constants.
            API_ADDRESS;
        Constants.API_TRANSLATE = Constants.
            API_ROOT + "/translate";
        Constants.API_SUGGEST = Constants.API_ROOT
            + "/suggest";
    }
}

@Subscribe
public void appendTranslation(EventsManager.
      RequestTranslationEvent e){
    if(e.status == Constants.STATUS_CODE_ACCEPTED
        ){
        if(language_mode == 0) {
            translations.add(e.input);
            locations.add(e.location);
        }
        else if(language_mode == 1){
            translations2.add(e.input);
            locations2.add(e.location);
        }
        clearExcess();
    }
    else{
        Toast.makeText(this, "Invalid Server Address or an
            unknown error.", Toast.LENGTH_SHORT).
            show();
        inputView.setText(e.input);
        inputView.setSelection(e.input.length());
    }
}

private void getTranslation(String location){
    apiManager.getTranslation(location);
}

@Subscribe
public void saveTranslation(EventsManager.
      GetTraslationEvent e){
    if(e.status == Constants.STATUS_CODE_SUCCESS){
        if (e.state.equals("SUCCESS")) {
            createInstantDialogWithSuggest("Translation", e
                .translation);
        }
        else if(e.state.equals("PENDING")){
            createInstantDialog(e.state, "Your input is
                already added to the queue.");
        }
        else if(e.state.equals("TRANSLATING")){
            createInstantDialog(e.state, "Your input is
                being translated.");
        }
        else if(e.state.equals("FAILED")){
            createInstantDialog(e.state, "Something went
                wrong. Sorry for the inconvenience.");
        }
    }
    else{
        Toast.makeText(this, "Invalid Server Address or an
            unknown error.", Toast.LENGTH_SHORT).
            show();
    }
}

private void clearExcess(){
    if(language_mode == 0) {
        if (translations.size() > HISTORY_COUNT) {
            while(translations.size() > HISTORY_COUNT
                ) {
                translations.remove(0);
                locations.remove(0);
            }
        }

        if (translations != null) {
            TranslationsAdapter adapter = new
                TranslationsAdapter(translations);
            translationListView.setAdapter(adapter);

            ItemClickSupport.addTo(translationListView).
                setOnItemClickListener(new
                ItemClickSupport.OnItemClickListener() {
                @Override
                public void onItemClicked(RecyclerView
                    recyclerView, int position, View v) {
                    current_click_input = translations.get(
                        position);
                    getTranslation(locations.get(position));
                }
            });
        }
    }
    else if(language_mode == 1) {
        if (translations2.size() > HISTORY_COUNT) {
            while(translations2.size() >
```

```
                    HISTORY_COUNT) {
                translations2.remove(0);
                locations2.remove(0);
            }
        }

        if (translations2 != null) {
            TranslationsAdapter adapter = new
                    TranslationsAdapter(translations2);
            translationListView.setAdapter(adapter);

            ItemClickSupport.addTo(translationListView).
                    setOnItemClickListener(new
                    ItemClickSupport.OnItemClickListener() {
                @Override
                public void onItemClicked(RecyclerView
                        recyclerView, int position, View v) {
                    current_click_input = translations2.get(
                            position);
                    getTranslation(locations2.get(position))
                            ;
                }
            });
        }
    }
}

public void createInstantDialog(String title, String
      message){
    AlertDialog.Builder builder = new AlertDialog.Builder(
          this, R.style.AppCompatAlertDialogStyle);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.setCancelable(true);
    builder.setPositiveButton(android.R.string.ok,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                        int id) {
                    dialog.cancel();
                }
            });
    AlertDialog alert = builder.create();
    alert.show();
}

public void createInstantDialogWithSuggest(String title,
      String message){
    final String translation = message;
    final AlertDialog.Builder builder = new AlertDialog.
          Builder(this, R.style.AppCompatAlertDialogStyle)
          ;
    builder.setTitle(title);
    builder.setMessage(message);
    builder.setCancelable(true);
    builder.setNegativeButton(android.R.string.ok,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                        int id) {
                    dialog.cancel();
                }
            });
    builder.setPositiveButton(R.string.suggest,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                        int id) {
                    startSuggest(translation);
                    dialog.cancel();
                }
            });

    AlertDialog alert = builder.create();
    alert.show();
}

private void startSuggest(String translation){
    SuggestActivity_.IntentBuilder_ intent =
          SuggestActivity_.intent(this);
    intent.extra("language_mode", language_mode);
    intent.extra("translation", translation);
    intent.extra("input", current_click_input).start();
}

@Click(R.id.ib_info)
void showInfo(){
    InfoActivity_.IntentBuilder_ intent = InfoActivity_.
          intent(this);
    intent.start();
}

@Click(R.id.ib_settings)
void showSettings(){
    settingsView.bringToFront();
    settingsView.loadValues();
    settingsView.setVisibility(View.VISIBLE);
}

@Click(R.id.ib_switch)
void switchLanguage() {
    mCurrRotation %= 360;
    float fromRotation = mCurrRotation;
    float toRotation = mCurrRotation += 360;

    final RotateAnimation rotateAnim = new
          RotateAnimation(
            fromRotation, toRotation, switchView.getWidth
                  ()/2, switchView.getHeight()/2);

    rotateAnim.setDuration(250); // Use 0 ms to rotate
          instantly
    rotateAnim.setFillAfter(true); // Must be true or the
          animation will reset

    switchView.startAnimation(rotateAnim);

    if(language_mode == 0){
        language_mode = 1;
        Toast.makeText(this, "English to Filipino", Toast.
              LENGTH_SHORT).show();
        languageOneView.setText(getResources().getString(
              R.string.language_2));
        languageTwoView.setText(getResources().getString(
              R.string.language_1));
        inputView.setHint(R.string.hint_english);
    }
    else{
        language_mode = 0;
        Toast.makeText(this, "Filipino to English", Toast.
              LENGTH_SHORT).show();
        languageOneView.setText(getResources().getString(
              R.string.language_1));
        languageTwoView.setText(getResources().getString(
              R.string.language_2));
        inputView.setHint(R.string.hint_filipino);
    }
    clearExcess();
}
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\adapters\TranslationsAdapter.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.adapters;

import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.ArrayList;

import agtacorda.dpsm.cas.upm.neurotranslate.R;
import agtacorda.dpsm.cas.upm.neurotranslate.views.
      TranslatedItemView_;

/**
 * Created by Alfred on 1/22/2017.
 */

public class TranslationsAdapter extends RecyclerView.Adapter
      <TranslationsAdapter.ViewHolder>{

    private ArrayList<String> translations;

    public static class ViewHolder extends RecyclerView.
          ViewHolder {
        protected TextView vTranslatedItem;

        public ViewHolder(View v) {
            super(v);
            vTranslatedItem = (TextView) v.findViewById(R.id.
                  tv_translated_item);
        }
    }

    public TranslationsAdapter(ArrayList<String> translations)
          {
        this.translations = translations;
    }

    @Override
    public int getItemCount() {
        return translations.size();
    }
```

```
    @Override
    public void onBindViewHolder(TranslationsAdapter.
            ViewHolder viewHolder, int i) {
        viewHolder.vTranslatedItem.setText(translations.get(i));
    }

    @Override
    public TranslationsAdapter.ViewHolder onCreateViewHolder
            (ViewGroup viewGroup, int i) {
        View itemView = TranslatedItemView_
                .build(viewGroup.getContext());

        return new TranslationsAdapter.ViewHolder(itemView);
    }
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\extra\ItemClickSupport.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.extra;

import android.support.v7.widget.RecyclerView;
import android.view.View;

import agtacorda.dpsm.cas.upm.neurotranslate.R;

/**
 * Created by Alfred on 1/22/2017.
 */

public class ItemClickSupport {
    private final RecyclerView mRecyclerView;
    private OnItemClickListener mOnItemClickListener;
    private OnItemLongClickListener
            mOnItemLongClickListener;
    private View.OnClickListener mOnClickListener = new
            View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mOnItemClickListener != null) {
                RecyclerView.ViewHolder holder =
                        mRecyclerView.getChildViewHolder(v);
                mOnItemClickListener.onItemClicked(
                        mRecyclerView, holder.getAdapterPosition
                        (), v);
            }
        }
    };
    private View.OnLongClickListener mOnLongClickListener =
            new View.OnLongClickListener() {
        @Override
        public boolean onLongClick(View v) {
            if (mOnItemLongClickListener != null) {
                RecyclerView.ViewHolder holder =
                        mRecyclerView.getChildViewHolder(v);
                return mOnItemLongClickListener.
                        onItemLongClicked(mRecyclerView, holder
                        .getAdapterPosition(), v);
            }
            return false;
        }
    };
    private RecyclerView.OnChildAttachStateChangeListener
            mAttachListener
            = new RecyclerView.
                    OnChildAttachStateChangeListener() {
        @Override
        public void onChildViewAttachedToWindow(View view)
                {
            if (mOnItemClickListener != null) {
                view.setOnClickListener(mOnClickListener);
            }
            if (mOnItemLongClickListener != null) {
                view.setOnLongClickListener(
                        mOnLongClickListener);
            }
        }

        @Override
        public void onChildViewDetachedFromWindow(View
                view) {

        }
    };

    private ItemClickSupport(RecyclerView recyclerView) {
        mRecyclerView = recyclerView;
        mRecyclerView.setTag(R.id.item_click_support, this);
        mRecyclerView.addOnChildAttachStateChangeListener(
                mAttachListener);
    }

    public static ItemClickSupport addTo(RecyclerView view) {
        ItemClickSupport support = (ItemClickSupport) view.
                getTag(R.id.item_click_support);
        if (support == null) {
            support = new ItemClickSupport(view);
        }
        return support;
    }

    public static ItemClickSupport removeFrom(RecyclerView
            view) {
        ItemClickSupport support = (ItemClickSupport) view.
                getTag(R.id.item_click_support);
        if (support != null) {
            support.detach(view);
        }
        return support;
    }

    public ItemClickSupport setOnItemClickListener(
            OnItemClickListener listener) {
        mOnItemClickListener = listener;
        return this;
    }

    public ItemClickSupport setOnItemLongClickListener(
            OnItemLongClickListener listener) {
        mOnItemLongClickListener = listener;
        return this;
    }

    private void detach(RecyclerView view) {
        view.removeOnChildAttachStateChangeListener(
                mAttachListener);
        view.setTag(R.id.item_click_support, null);
    }

    public interface OnItemClickListener {

        void onItemClicked(RecyclerView recyclerView, int
                position, View v);
    }

    public interface OnItemLongClickListener {

        boolean onItemLongClicked(RecyclerView recyclerView,
                int position, View v);
    }
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\managers\APIManager.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.managers;

import android.util.Log;

import com.fasterxml.jackson.core.JsonFactory;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.ObjectReader;

import org.androidannotations.annotations.Background;
import org.androidannotations.annotations.Bean;
import org.androidannotations.annotations.EBean;
import org.androidannotations.annotations.UiThread;
import org.json.JSONObject;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.Cookie;
import okhttp3.CookieJar;
import okhttp3.HttpUrl;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

/**
 * Created by Alfred on 3/13/2017.
 */

@EBean(scope = EBean.Scope.Singleton)
public class APIManager {
```

```
private ObjectMapper mapper;
private ObjectReader reader;
private JsonFactory factory;

public static final MediaType JSON = MediaType.parse("
      application/json; charset=utf-8");
public static final MediaType IMAGE = MediaType.parse
      ("image/png; charset=utf-8");
public static final MediaType VOICE = MediaType.parse
      ("audio/amr; charset=utf-8");

// should be a singleton
OkHttpClient client;

public APIManager() {
    mapper = new ObjectMapper();
    reader = mapper.reader();
    factory = mapper.getFactory();

    OkHttpClient.Builder builder = new OkHttpClient.
          Builder();
    builder.cookieJar(new MyCookieJar());
    client = builder.build();
}


@Bean
OttoBus bus;

@Background
public void requestTranslation(JSONObject requestData,
      String input) {//post
    HttpUrl.Builder urlBuilder = HttpUrl.parse(Constants.
          API_TRANSLATE).newBuilder();
    //urlBuilder.addQueryParameter("v", "1.0");
    String url = urlBuilder.build().toString();
    final String input2 = input;

    RequestBody body = RequestBody.create(JSON,
          requestData.toString());

    Request request = new Request.Builder()
            .url(url)
            .post(body)
            .build();

    client.newCall(request).enqueue(
            new Callback() {
                @Override
                public void onFailure(Call call,
                      IOException e) {
                    e.printStackTrace();
                    sendRequestTranslationResult("", 600,
                          input2); //600 is not documented
                          in the REST API
                    Log.i("INFO","request Failure");
                }

                @Override
                public void onResponse(Call call, final
                      Response response) throws
                      IOException {
                    if (!response.isSuccessful()) {
                        sendRequestTranslationResult("",
                              response.code(), input2);
                        throw new IOException("
                              Unexpected code " + response
                              );
                    }
                    else{
                        Log.i("INFO",response.toString());

                        if(response.code() == Constants.
                              STATUS_CODE_ACCEPTED)
                              {
                            String responseData = response.
                                  body().string();
                            ObjectMapper mapper = new
                                  ObjectMapper();

                            JsonNode rootNode = mapper.
                                  readValue(responseData,
                                  JsonNode.class);
                            String location = rootNode.get
                                  ("location").asText();

                            sendRequestTranslationResult(
                                  location, response.code(),
                                  input2);
                        }
                        else{
                            sendRequestTranslationResult
                                  ("", response.code(),
                                  input2);
                        }
                    }
                }
            });
}

@UiThread
void sendRequestTranslationResult(String location, int
      status, String input) {
    bus.post(new EventsManager.RequestTranslationEvent(
          location, status, input));
    Log.i("INFO", "Sending rtranslate status(" + status +
          ")!! using Ottobus");
    Log.i("INFO", "location: (" + location + ")!! using
          Ottobus");
}

@Background
public void getTranslation(String location) {//get
    HttpUrl.Builder urlBuilder = HttpUrl.parse(Constants.
          API_ROOT + location).newBuilder();
    //urlBuilder.addQueryParameter("v", "1.0");
    String url = urlBuilder.build().toString();

    Request request = new Request.Builder()
            .url(url)
            .build();

    client.newCall(request).enqueue(
            new Callback() {
                @Override
                public void onFailure(Call call,
                      IOException e) {
                    e.printStackTrace();
                    sendTranslationResult("", "", 600);
                          //600 is not documented in the
                          REST API
                    Log.i("INFO","request Failure");
                }

                @Override
                public void onResponse(Call call, final
                      Response response) throws
                      IOException {
                    if (!response.isSuccessful()) {
                        sendTranslationResult("", "FAILED
                              ", response.code());
                        throw new IOException("
                              Unexpected code " + response
                              );
                    }
                    else{
                        Log.i("INFO",response.toString());

                        if(response.code() == Constants.
                              STATUS_CODE_SUCCESS) {
                            String responseData = response.
                                  body().string();
                            ObjectMapper mapper = new
                                  ObjectMapper();

                            JsonNode rootNode = mapper.
                                  readValue(responseData,
                                  JsonNode.class);
                            String translation = rootNode.
                                  get("translation").asText
                                  ();
                            String state = rootNode.get("
                                  state").asText();

                            sendTranslationResult(
                                  translation, state,
                                  response.code());
                        }
                        else{
                            sendTranslationResult("", "
                                  ERROR", response.code()
                                  );
                        }
                    }
                }
            });
}

@UiThread
void sendTranslationResult(String translation, String state,
      int status) {
    bus.post(new EventsManager.GetTraslationEvent(
          translation, state, status));
    Log.i("INFO", "Sending gstatus(" + status + ")!! using
```

```
            Ottobus");
        Log.i("INFO", "translation: (" + translation + ")!!
            using Ottobus");
    }

    @Background
    public void suggestTranslation(JSONObject requestData)
        {//post
        HttpUrl.Builder urlBuilder = HttpUrl.parse(Constants.
            API_SUGGEST).newBuilder();
        //urlBuilder.addQueryParameter("v", "1.0");
        String url = urlBuilder.build().toString();

        RequestBody body = RequestBody.create(JSON,
            requestData.toString());

        Request request = new Request.Builder()
                .url(url)
                .post(body)
                .build();

        client.newCall(request).enqueue(
                new Callback() {
                    @Override
                    public void onFailure(Call call,
                            IOException e) {
                        e.printStackTrace();
                        sendSuggestTranslationResult(600);
                                //600 is not documented in the
                                REST API
                    }

                    @Override
                    public void onResponse(Call call, final
                            Response response) throws
                            IOException {
                        if (!response.isSuccessful()) {
                            sendSuggestTranslationResult(
                                    response.code());
                            throw new IOException("
                                    Unexpected code " + response
                                    );
                        }
                        else{
                            Log.i("INFO",response.toString());

                            if(response.code() == Constants.
                                    STATUS_CODE_SUCCESS) {
                                String responseData = response.
                                        body().string();
                                ObjectMapper mapper = new
                                        ObjectMapper();

                                sendSuggestTranslationResult(
                                        response.code());
                            }
                            else{
                                sendSuggestTranslationResult(
                                        response.code());
                            }
                        }
                    }
                });
    }

    @UiThread
    void sendSuggestTranslationResult(int status) {
        bus.post(new EventsManager.SuggestTranslationEvent(
                status));
        Log.i("INFO", "Sending suggest status(" + status + ")!!
                using Ottobus");
    }

    class MyCookieJar implements CookieJar {

        private List<Cookie> cookies;

        @Override
        public void saveFromResponse(HttpUrl url, List<Cookie
                > cookies) {
            this.cookies = cookies;
        }

        @Override
        public List<Cookie> loadForRequest(HttpUrl url) {
            if (cookies != null)
                return cookies;
            return new ArrayList<>();

        }
    }
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\managers\Constants.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.managers;

/**
 * Created by Alfred on 4/10/2017.
 */

public class Constants {
    public static String API_ADDRESS = "192.168.137.1";

    public static String API_ROOT = "http://" +
        API_ADDRESS;

    public static String API_TRANSLATE = API_ROOT + "/
        translate";

    public static String API_SUGGEST = API_ROOT + "/
        suggest";

    public static final int STATUS_CODE_SUCCESS = 200;

    public static final int STATUS_CODE_ACCEPTED =
        202;
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\managers\EventsManager.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.managers;

/**
 * Created by Alfred on 4/10/2017.
 */

public class EventsManager {
    public EventsManager(){}

    public static class RequestTranslationEvent{
        public String location;
        public int status;
        public String input;

        RequestTranslationEvent(String location, int status,
                String input){
            this.location = location;
            this.status = status;
            this.input = input;
        }
    }

    public static class GetTraslationEvent{
        public String translation;
        public String state;
        public int status;

        GetTraslationEvent(String translation, String state, int
                status){
            this.translation = translation;
            this.state = state;
            this.status = status;
        }
    }

    public static class SuggestTranslationEvent{
        public int status;

        SuggestTranslationEvent(int status){
            this.status = status;
        }
    }

    public static class PreferencesUpdated{
    }
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\managers\OttoBus.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.managers;

import com.squareup.otto.Bus;

import org.androidannotations.annotations.EBean;

/**
 * Created by Alfred on 4/10/2017.
```

```
 */

@EBean(scope = EBean.Scope.Singleton)
public class OttoBus extends Bus {

}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\views\SettingsView.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.views;

import android.content.Context;
import android.content.SharedPreferences;
import android.util.AttributeSet;
import android.view.View;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.Toast;

import org.androidannotations.annotations.AfterViews;
import org.androidannotations.annotations.Bean;
import org.androidannotations.annotations.Click;
import org.androidannotations.annotations.EViewGroup;
import org.androidannotations.annotations.ViewById;

import agtacorda.dpsm.cas.upm.neurotranslate.R;
import agtacorda.dpsm.cas.upm.neurotranslate.activities.
        TranslatorActivity;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.
        Constants;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.
        EventsManager;
import agtacorda.dpsm.cas.upm.neurotranslate.managers.
        OttoBus;

import static android.content.Context.MODE_PRIVATE;

/**
 * Created by Alfred on 5/20/2017.
 */
@EViewGroup(R.layout.view_settings)
public class SettingsView extends FrameLayout{
    @ViewById(R.id.et_address)
    EditText address;

    @ViewById(R.id.et_max)
    EditText max;

    @Bean
    OttoBus bus;

    public static final String preferences = "
            neurotranslatePreferences" ;

    SharedPreferences prefs = getContext().
            getSharedPreferences(preferences,MODE_PRIVATE);
    SharedPreferences.Editor editor = prefs.edit();

    public SettingsView(Context context, AttributeSet attrs){
        super(context, attrs);
    }

    @AfterViews
    void initialize (){
        loadValues();
    }

    public void loadValues(){
        if(prefs.contains("address")) {
            address.setText(prefs.getString("address",
                    Constants.API_ADDRESS));
        }
        else{
            address.setText(Constants.API_ADDRESS);
        }

        if(prefs.contains("max")) {
            max.setText(prefs.getString("max", "10"));
        }
        else{
            max.setText("10");
        }
    }

    @Click(R.id.save)
    void save(){
        String address_save = address.getText().toString();
        String max_save = max.getText().toString();

        if(Integer.parseInt(max_save) < 1){
            Toast.makeText(getContext(), "Invalid Max
                    Translations", Toast.LENGTH_SHORT).show
                    ();
        }
        else {
            editor.putString("address", address_save);
            editor.putString("max", max_save);
            editor.commit();
            editor.apply();
            bus.post(new EventsManager.PreferencesUpdated());
            setVisibility (View.GONE);
            Toast.makeText(getContext(), "Settings Saved",
                    Toast.LENGTH_SHORT).show();
        }
    }

    @Click(R.id.cancel)
    void close(){
        setVisibility (View.GONE);
    }
}
```

File: NeuroTranslate\app\src\main\java\agtacorda\dpsm\cas\upm\neurotranslate\views\TranslatedItemView.java

```
package agtacorda.dpsm.cas.upm.neurotranslate.views;

import android.content.Context;
import android.widget.RelativeLayout;
import android.widget.TextView;

import org.androidannotations.annotations.EViewGroup;
import org.androidannotations.annotations.ViewById;

import agtacorda.dpsm.cas.upm.neurotranslate.R;

/**
 * Created by Alfred on 1/22/2017.
 */

@EViewGroup(R.layout.view_translated_item)
public class TranslatedItemView extends RelativeLayout{

    @ViewById(R.id.tv_translated_item)
    TextView translatedItemView;

    public TranslatedItemView(Context context){
        super(context);
    }

    public void setTranslatedItemText(String text){
        translatedItemView.setText(text);
    }
}
```

File: NeuroTranslate\app\src\main\res\drawable\color_cursor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/
        android" >
    <size android:width="1dp" />
    <solid android:color="@color/colorAccent" />
</shape>
```

File: NeuroTranslate\app\src\main\res\drawable\et_rounded.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/
        android" >
    <solid android:color="#FFFFFF" />
    <stroke
        android:width="1dp"
        android:color="#808080" />
    <corners
        android:radius="3dp"/>
</shape>
```

File: NeuroTranslate\app\src\main\res\drawable\shadow.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/
        android" android:shape="rectangle">
    <solid android:color="#FFFFFF"/>
    <stroke android:width="1dp" android:color="#808080"/>
    <corners android:radius="2dp"/>
</shape>
```

File: NeuroTranslate\app\src\main\res\drawable\tv_rounded.

xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/
    android" >
    <solid android:color="#FFFFFF" />
    <stroke
        android:width="1dp"
        android:color="@color/colorAccent" />
    <corners
        android:radius="3dp"/>
</shape>
```

File: NeuroTranslate\app\src\main\res\drawable-anydpi\
ic_arrow_back.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/
    android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0">
    <path
        android: fillColor ="#FFFFFF"
        android:pathData="M20,11H7.83l5.59,-5.59L12,4l-8,8
            8,8 1.41,-1.41L7.83,13H20v-2z"/>
</vector>
```

File: NeuroTranslate\app\src\main\res\drawable-anydpi\
ic_info_outline.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/
    android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0">
    <path
        android: fillColor ="#FFFFFF"
        android:pathData="M11,17h2v-6h-2v6zM12,2C6.48,2
            2,6.48 2,12s4.48,10 10,10 10,-4.48 10,-10S17.52,2
            12,2zM12,20c-4.41,0 -8,-3.59 -8,-8s3.59,-8
            8,-8 8,3.59 8,8 -3.59,8 -8,8zM11,9h2L13,7h-2
            v2z"/>
</vector>
```

File: NeuroTranslate\app\src\main\res\drawable-anydpi\
ic_settings.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/
    android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0">
    <path
        android: fillColor ="#FFFFFF"
        android:pathData="M19.43,12.98c0.04,-0.32 0.07,-0.64
            0.07,-0.98s-0.03,-0.66 -0.07,-0.98l2.11,-1.65c0
            .19,-0.15 0.24,-0.42 0.12,-0.64l-2,-3.46c
            -0.12,-0.22 -0.39,-0.3 -0.61,-0.22l-2.49,1c
            -0.52,-0.4 -1.08,-0.73 -1.69,-0.98l-0.38,-2.65
            C14.46,2.18 14.25,2 14,2h-4c-0.25,0 -0.46,0.18
            -0.49,0.42l-0.38,2.65c-0.61,0.25 -1.17,0.59
            -1.69,0.98l-2.49,-1c-0.23,-0.09 -0.49,0
            -0.61,0.22l-2,3.46c-0.13,0.22 -0.07,0.49
            0.12,0.64l2.11,1.65c-0.04,0.32 -0.07,0.65
            -0.07,0.98s0.03,0.66 0.07,0.98l-2.11,1.65c
            -0.19,0.15 -0.24,0.42 -0.12,0.64l2,3.46c0.12,0.22
            0.39,0.3 0.61,0.22l2.49,-1c0.52,0.4 1.08,0.73
            1.69,0.98l0.38,2.65c0.03,0.24 0.24,0.42 0.49,0.42
            h4c0.25,0 0.46,-0.18 0.49,-0.42l0.38,-2.65c0
            .61,-0.25 1.17,-0.59 1.69,-0.98l2.49,1c0.23,0.09
            0.49,0 0.61,-0.22l2,-3.46c0.12,-0.22 0.07,-0.49
            -0.12,-0.64l-2.11,-1.65zM12,15.5c-1.93,0
            -3.5,-1.57 -3.5,-3.5s1.57,-3.5 3.5,-3.5 3.5,1.57
            3.5,3.5 -1.57,3.5 -3.5,3.5z"/>
</vector>
```

File: NeuroTranslate\app\src\main\res\layout\activity_info.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F7911E"
    android:id="@+id/activity_translator"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="#FFFFFF">

        <android.support.v7.widget.Toolbar xmlns:app="http://
            schemas.android.com/apk/res-auto"
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="@color/colorPrimary"
            android:theme="@style/ThemeOverlay.AppCompat.
                ActionBar">

            <ImageButton
                android:id="@+id/ib_back"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:src="@drawable/ic_arrow_back"
                android:background="@color/colorPrimary"
                android:layout_gravity="start"
                android:layout_marginLeft="13dp"
                android:layout_marginStart="13dp"/>

            <TextView
                android:id="@+id/tv_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/about"
                android:textColor="@android:color/white"
                android:textSize="21sp" />
        </android.support.v7.widget.Toolbar>

        <View
            android:layout_width="match_parent"
            android:layout_height="3dp"
            android:layout_alignParentBottom="true"
            android:background="@color/colorAccent" />

    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/content_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#F9F9F9"
        android: scrollbars ="vertical">

        <TextView
            android:id="@+id/tv_info1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/info_1"
            android:layout_marginTop="20dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp"
            android:textColor="@android:color/black"
            android:textSize="18sp" />

        <TextView
            android:id="@+id/tv_info2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/tv_info1"
            android:text="@string/info_2"
            android:layout_marginTop="20dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp"
            android:textColor="@android:color/black"
            android:textSize="18sp" />

        <TextView
            android:id="@+id/tv_info3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/tv_info2"
            android:text="@string/info_3"
            android:layout_marginTop="20dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp"
            android:textColor="@android:color/black"
            android:textSize="18sp" />
    </RelativeLayout>
</LinearLayout>
```

File: NeuroTranslate\app\src\main\res\layout\activity_suggest.
xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
      apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F7911E"
    android:id="@+id/activity_translator"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="#FFFFFF">

        <android.support.v7.widget.Toolbar xmlns:app="http://
              schemas.android.com/apk/res-auto"
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="@color/colorPrimary"
            android:theme="@style/ThemeOverlay.AppCompat.
                  ActionBar">

            <ImageButton
                android:id="@+id/ib_back"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:src="@drawable/ic_arrow_back"
                android:background="@color/colorPrimary"
                android:layout_gravity="start"
                android:layout_marginLeft="13dp"
                android:layout_marginStart="13dp" />

            <TextView
                android:id="@+id/tv_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/suggest"
                android:textColor="@android:color/white"
                android:textSize="21sp" />
        </android.support.v7.widget.Toolbar>

        <View
            android:layout_width="match_parent"
            android:layout_height="3dp"
            android:layout_alignParentBottom="true"
            android:background="@color/colorAccent" />

    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/content_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#F9F9F9"
        android:scrollbars="vertical">

        <EditText
            android:id="@+id/et_suggestion"
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:padding="10dp"
            android:gravity="start"
            android:maxLines="6"
            android:background="@drawable/tv_rounded"
            android:layout_marginTop="10dp"
            android:layout_marginRight="10dp"
            android:layout_marginLeft="10dp"
            android:layout_marginBottom="15dp"
            android:textSize="18sp"
            android:hint="@string/hint_suggestion"
            android:textCursorDrawable="@drawable/
                  color_cursor"
            android:inputType="textMultiLine"
            android:scrollbars="vertical" />

        <TextView
            android:id="@+id/tv_input"
            android:layout_below="@id/et_suggestion"
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:padding="10dp"
            android:gravity="start"
            android:maxLines="6"
            android:background="@drawable/et_rounded"
            android:layout_marginRight="10dp"
            android:layout_marginLeft="10dp"
            android:layout_marginBottom="15dp"
            android:textSize="18sp"
            android:scrollbars="vertical" />
    </RelativeLayout>
</LinearLayout>
```

File: NeuroTranslate\app\src\main\res\layout\
activity_translator.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
      apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F7911E"
    android:id="@+id/activity_translator"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="#FFFFFF">

        <android.support.v7.widget.Toolbar xmlns:app="http://
              schemas.android.com/apk/res-auto"
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="@color/colorPrimary"
            android:theme="@style/ThemeOverlay.AppCompat.
                  ActionBar">

            <TextView
                android:id="@+id/tv_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="start"
                android:layout_marginRight="2dp"
                android:layout_marginEnd="2dp"
                android:text="@string/app_name_1"
                android:textColor="@android:color/white"
                android:textSize="21sp" />

            <TextView
                android:id="@+id/tv_title2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="start"
                android:text="@string/app_name_2"
                android:textColor="@color/colorAccent"
                android:textSize="20sp" />


            <ImageButton
                android:id="@+id/ib_info"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:src="@drawable/ic_info_outline"
                android:background="@color/colorPrimary"
                android:layout_gravity="end"
                android:layout_marginRight="13dp"
                android:layout_marginEnd="13dp"/>


            <ImageButton
                android:id="@+id/ib_settings"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:src="@drawable/ic_settings"
                android:background="@color/colorPrimary"
                android:contentDescription="@string/
                      info_description"
                android:layout_gravity="end"
                android:layout_marginRight="17dp"
                android:layout_marginEnd="17dp" />

        </android.support.v7.widget.Toolbar>

        <View
            android:layout_width="match_parent"
            android:layout_height="3dp"
            android:layout_alignParentBottom="true"
            android:background="@color/colorAccent" />

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="#F9F9F9"
        android:orientation="horizontal">

        <TextView
```

```
            android:id="@+id/tv_language_1"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:gravity="center"
            android:text="@string/language_1"
            android:textColor="@android:color/black"
            android:textSize="19sp"
            android:layout_toLeftOf="@+id/ib_switch"
            android:layout_toStartOf="@+id/ib_switch"
            android:layout_marginRight="60dp"
            android:layout_marginEnd="60dp"
            tools:ignore="RelativeOverlap" />

        <ImageButton
            android:id="@+id/ib_switch"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:src="@drawable/button_switch"
            android:background="#F9F9F9"
            android:layout_centerVertical="true"
            android:layout_centerHorizontal="true"
            android:contentDescription="@string/
                switch_description" />

        <TextView
            android:id="@+id/tv_language_2"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:gravity="center"
            android:text="@string/language_2"
            android:textColor="@android:color/black"
            android:textSize="19sp"
            android:layout_toRightOf="@+id/ib_switch"
            android:layout_toEndOf="@+id/ib_switch"
            android:layout_marginLeft="60dp"
            android:layout_marginStart="60dp" />

    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/content_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#F9F9F9"
        android:scrollbars="vertical">

        <EditText
            android:id="@+id/et_input"
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:padding="10dp"
            android:gravity="start"
            android:maxLines="6"
            android:background="@drawable/et_rounded"
            android:layout_marginRight="10dp"
            android:layout_marginLeft="10dp"
            android:layout_marginBottom="15dp"
            android:textSize="18sp"
            android:hint="@string/hint_filipino"
            android:textCursorDrawable="@drawable/
                color_cursor"
            android:inputType="textMultiLine"
            android:scrollbars="vertical" />

        <agtacorda.dpsm.cas.upm.neurotranslate.views.
            SettingsView_
            android:id="@+id/settings_view"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:clickable="true"
        />

        <android.support.v7.widget.RecyclerView
            android:id="@+id/rv_translation_list"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:overScrollMode="never"
            android:scrollbars="none"
            android:layout_below="@+id/et_input"
            android:layout_centerHorizontal="true" />
    </RelativeLayout>
</LinearLayout>
```

File: NeuroTranslate\app\src\main\res\layout\view_settings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<merge xmlns:android="http://schemas.android.com/apk/res/
    android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/shadow"
        android:layout_marginRight="10dp"
        android:layout_marginEnd="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginStart="10dp">

        <TextView
            android:id="@+id/tv_label0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:layout_marginStart="15dp"
            android:layout_marginLeft="15dp"
            android:layout_marginBottom="10dp"
            android:text="@string/settings"
            android:textColor="#000000"
            android:textSize="21sp" />

        <View
            android:id="@+id/divider"
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:layout_marginBottom="15dp"
            android:background="@color/colorAccent"
            android:layout_below="@id/tv_label0" />

        <TextView
            android:id="@+id/tv_label1"
            android:layout_below="@id/divider"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="15dp"
            android:layout_marginLeft="15dp"
            android:layout_marginBottom="10dp"
            android:text="@string/hint_address"
            android:textSize="19sp" />

        <EditText
            android:id="@+id/et_address"
            android:layout_below="@+id/tv_label1"
            android:layout_width="match_parent"
            android:layout_height="35dp"
            android:padding="5dp"
            android:layout_marginStart="15dp"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp"
            android:layout_marginEnd="15dp"
            android:layout_marginBottom="15dp"
            android:background="@drawable/et_rounded"
            android:inputType="text"
            android:textSize="15sp" />

        <TextView
            android:id="@+id/tv_label2"
            android:layout_below="@+id/et_address"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="15dp"
            android:layout_marginLeft="15dp"
            android:layout_marginBottom="10dp"
            android:text="@string/hint_max"
            android:textSize="19sp" />

        <EditText
            android:id="@+id/et_max"
            android:layout_below="@+id/tv_label2"
            android:layout_width="match_parent"
            android:layout_height="35dp"
            android:padding="5dp"
            android:layout_marginStart="15dp"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp"
            android:layout_marginEnd="15dp"
            android:layout_marginBottom="15dp"
            android:background="@drawable/et_rounded"
            android:inputType="number"
            android:textSize="15sp" />

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/et_max"
            android:orientation="horizontal"
            android:layout_marginBottom="5dp">
```

```
            <Button
                android:id="@+id/cancel"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@null"
                android:text="@string/cancel"
                android:layout_gravity="end"
                android:textColor="#000000"
                android:layout_alignParentLeft="true"
                android:layout_marginBottom="15dp" />

            <Button
                android:id="@+id/save"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@null"
                android:text="@string/save"
                android:layout_gravity="end"
                android:textColor="@color/colorAccent"
                android:layout_alignParentRight="true"
                android:layout_marginBottom="15dp" />
        </RelativeLayout>
    </RelativeLayout>
</merge>
```

File: NeuroTranslate\app\src\main\res\layout\view_translated_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<merge xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/tv_translated_item"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:padding="10dp"
        android:gravity="start"
        android:maxLines="6"
        android:background="@drawable/tv_rounded"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="15dp"
        android:textSize="18sp" />

    <!-- Fix for versions with marginBottom not working. -->
    <View
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_below="@id/tv_translated_item" />

</merge>
```

File: NeuroTranslate\app\src\main\res\values\colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <!--<color name="colorPrimary">#0066CC</color> original-->
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FFD700</color>
</resources>
```

File: NeuroTranslate\app\src\main\res\values\dimens.xml

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

File: NeuroTranslate\app\src\main\res\values\ids.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <item name="item_click_support" type="id" />
</resources>
```

File: NeuroTranslate\app\src\main\res\values\strings.xml

```
<resources>
    <string name="app_name">Neuro Translate</string>
    <string name="app_name_1">Neuro </string>
    <string name="app_name_2">Translate</string>
    <string name="suggest">Suggest</string>
    <string name="about">About</string>
    <string name="language_1">Filipino</string>
    <string name="language_2">English</string>
    <string name="info_description">Guide</string>
    <string name="switch_description">Switch Language</string>
    <string name="hint_filipino">Maglagay ng tekstong gustong isalin</string>
    <string name="hint_english">Input text to be translated</string>
    <string name="hint_suggestion">Your Suggestion Here.</string>
    <string name="hint_address">Server Address</string>
    <string name="hint_max">Max Translations</string>
    <string name="settings">Settings</string>
    <string name="save">Save</string>
    <string name="cancel">Cancel</string>
    <string name="info_1">Neuro Translate is created by Alfred Tacorda. It uses Neuro MT to serve translations .</string>
    <string name="info_2">To request a translation, enter your input in the uppermost textbox.</string>
    <string name="info_3">To get the translation, click the box that contains your input.</string>
</resources>
```

File: NeuroTranslate\app\src\main\res\values\styles.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

    <style name="AppCompatAlertDialogStyle" parent="Theme.AppCompat.Light.Dialog.Alert">
        <item name="colorAccent">@color/colorAccent</item>
        <item name="android:textColorPrimary">#000000</item>
        <item name="android:background">#FFFFFF</item>
    </style>

    <style name="Theme.AppCompat.Light.NoActionBar" parent="@style/Theme.AppCompat.Light">
        <item name="windowNoTitle">true</item>
        <item name="windowActionBar">false</item>
        <item name="android:windowFullscreen">false</item>
        <item name="android:windowContentOverlay">@null</item>
    </style>

</resources>
```

File: NeuroTranslate\app\src\main\res\values-w820dp\dimens.xml

```
<resources>
    <!-- Example customization of dimensions originally defined in res/values/dimens.xml
         (such as screen margins) for screens with more than 820dp of available width. This
         would include 7" and 10" devices in landscape (~960dp and ~1280dp respectively). -->
    <dimen name="activity_horizontal_margin">64dp</dimen>
</resources>
```

# XI. Acknowledgements

First of all, I would like to thank my parents for their unyielding and unquestioning support. Even if I do not ask for something, they magically know what I need.

I would also like to thank Sir Marvin Ignacio, for accepting me as his advisee, even if his initial slots were full. This allowed me to choose a topic that I want to pursue, and have more time to study it. His masterful plan to lighten our load for the first semester, helped me a lot in the proposal of this special problem. His lectures about deep learning, a subject not offered here in UP Manila, helped us understand this wonderful field.

The road to finishing this course was really rough, but with the help of my close friends: Andrei, Berwin, Elias, Ironnel, Jeffrey, Jethro, and Kier, we have reached our destination. To my fellow advisees: Damian, Elias, Jerome, Moses, Reinier, and Sheela, I enjoyed our pursuit for knowledge in this ever-changing field. And to my blockmates, thank you for understanding my shortcomings. I would also like to thank Neil, Adi, Jed, and the DPSM faculty, for leading our block and the computer science blocks before us. Their efforts lead to the steady improvement of the UP Manila ComSci community. I would also like to thank all the professors and lecturers who have taught me a lot of things. Your standards and beliefs were selectively chosen as a part of me now.

And finally, this project was not possible without the dataset provided by the Philippine-California Advanced Research Institutes(PCARI), through project entitled E-Participation 2.0: Connecting Diverse Philippine Populations for Disaster Risk Management with a Toolkit Integrating Text and Speech Analytics (project no. IIID 2015-07). I would like to thank Sir Nathaniel Oco and Ma'am Rachel Roxas of PCARI for their support, Dr. Samuel Grozman, for the funding of the 2x GTX 1070 PC(pagaspas), Rico Sennrich(the creator of Nematus), the Deep Learning community, the Python Community, and our Testers.