DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY
Academic Year 2018 - 2019

Neural Machine Translation Using Transformer and RNN Encoder-Decoder Pair Models

Authors:
Devashish Gaikwad 111608023
Atharva Jadhav 111608031
Third Year Information Technology

Contents

1 Abstract

The project aims to create English to German and English to French bidirectional language translation models based on the reference research papers mentioned below. We have chosen two models to implement - RNN Encoder Decoder Model and Transformer Model. The first deep learning neural network model architecture consists of encoder decoder pair of recurrent neural networks to achieve sequence to sequence phrase translation between the languages using a fixed common vector between the encoder and decoder blocks. The second neural network is a Transformer model which has self-attention layer and a feed forward neural network embedded into all encoder and decoder blocks, which have been stacked on the top of each other, this model also has better results and accuracy than the RNN Encoder-Decoder model and thus we have decided to implement this model into the end system

Accuracy of The models will be judged on the basis of BLEU (Bilingual Evaluation) score which is a benchmark for language translation models. After a successful training and testing of model, a web server to handle translation requests in text format will be created and hosted.

2 Model research papers in brief

2.1 Sequence to Sequence Learning with Neural Networks - Sutskever, Vinyals et al.

The encoder-decoder architecture for recurrent neural networks is the standard neural machine translation method that rivals and in some cases outperforms classical statistical machine translation methods. The Encoder-Decoder architecture with recurrent neural networks has become an effective and standard approach for both neural machine translation (NMT) and sequence-to-sequence (seq2seq) prediction in general. The key benefits of the approach are the ability to train a single end-to-end model directly on source and target sentences and the ability to handle variable length input and output sequences of text. This architecture is very new, having only been pioneered in 2014, although, has been adopted as the core technology inside Google's translate service.

we studied the neural machine translation model developed by Ilya Sutskever, et al. as described in their 2014 paper "Sequence to Sequence Learning with Neural Networks". We will refer to it as the "Sutskever NMT Model" and "RNN Encoder-Decoder Model", for lack of a better name. This is an important paper as it was one of the first to introduce the Encoder-Decoder model for machine translation and more generally sequence-to-sequence learning. It is an important model in the field of machine translation as it was one of the first neural machine translation systems to outperform a baseline statistical machine learning model on a large translation task.

An Encoder-Decoder architecture was developed where an input sequence was read in entirety and encoded to a fixed-length internal representation. A decoder network then used this internal representation to output words until the end of sequence token was reached. LSTM networks were used for both the encoder and decoder. The idea is to use one LSTM to read the input sequence, one timestep at a time, to obtain large fixed-dimensional vector representation, and then to use another LSTM to extract the output sequence from that vector

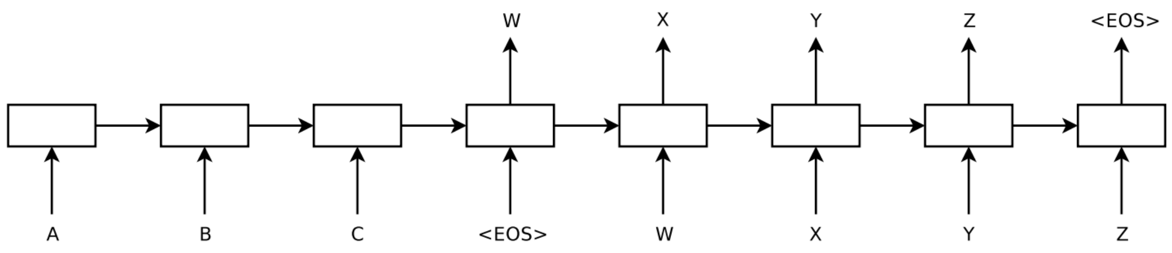


Figure 1: Architecture of Sutskever NMT Model

2.2 Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation - Kyunghyun Cho, et al

In this section, we will look at the neural machine translation system described by Kyunghyun Cho, et al. in their 2014 paper titled "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." We will refer to it as the "Cho NMT Model" model for lack of a better name. Importantly, the Cho Model is used only to score candidate translations and is not used directly for translation like the Sutskever model above. Although extensions to the work to better diagnose and improve the model do use it directly and alone for translation.

In this paper, Cho et al have proposed a novel neural network model called RNN Encoder– Decoder that consists of two recurrent neural networks (RNN). One RNN encodes a sequence of symbols into a fixed- length vector representation, and the other decodes the representation into another sequence of symbols. The encoder and decoder of the proposed model are jointly trained to maximize the conditional probability of a target sequence given a source sequence. The performance of a statistical machine translation system is empirically found to improve by using the conditional probabilities of phrase pairs computed by the RNN Encoder–Decoder as an additional feature in the existing log-linear model. Qualitatively, they have shown that the proposed model learns a semantically and syntactically meaningful representation of linguistic phrases.

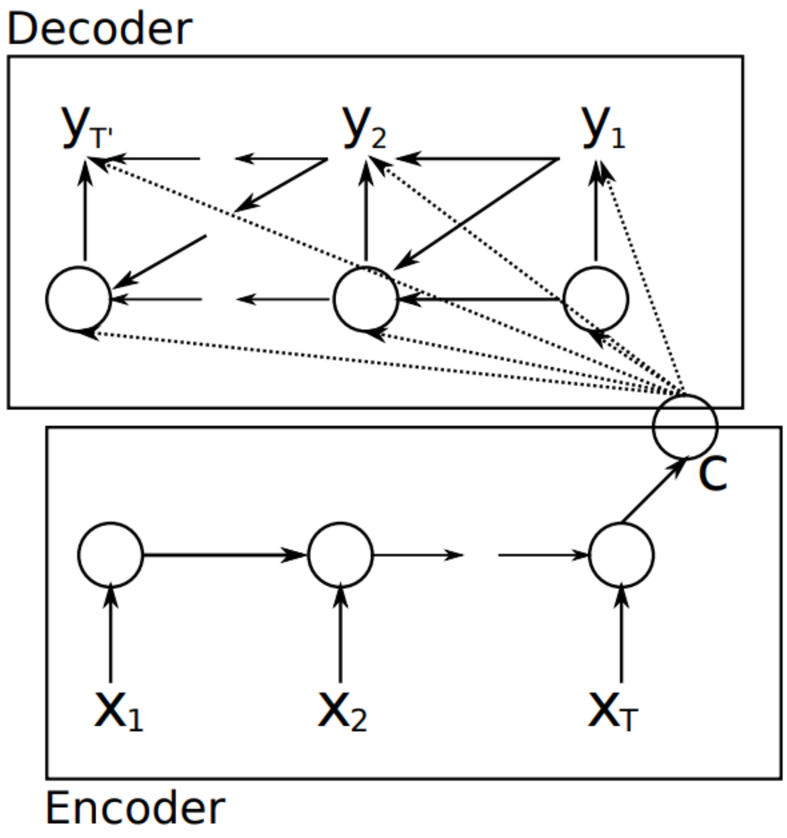


Figure 2: Architecture of Cho NMT Model

2.3 Transformers
Attention Is All You Need - waswani, shazeer, Polosukhin et al
Scaling Neural Machine Translation - Ott, Edunov et al

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states ht , as a function of the previous hidden state ht1 and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks and conditional computation, while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Transformers are the latest addition to NLP models. It uses self-attention and feed forward neural networks only while simplifying the architecture of the model and increasing performance and decreasing training time. Transformer model is based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations.

Architecture of model:

Encoder: The encoder suggested in research paper is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. They have also employed a residual connection around each of the two sub-layers, followed by layer normalization. That is, the output of each sub-layer is LayerNorm(x + Sublayer(x)), where Sublayer(x) is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension dmodel = 512.

Decoder: The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization. We also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i.

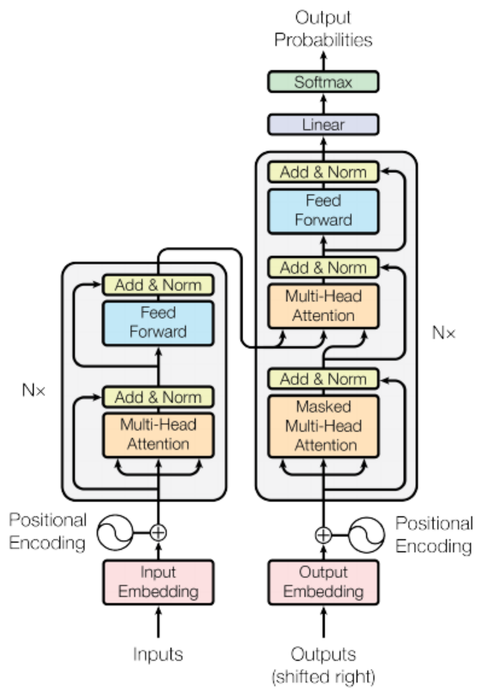


Figure 1: The Transformer - model architecture.

Figure 3: Architecture of Attention Model

Attention:

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. We call our particular attention ”Scaled Dot-Product Attention”. The input consists of queries and keys of dimension dk , and values of dimension dv . We compute the dot products of the query with all keys, divide each by square root of dk , and apply a softmax function to obtain the weights on the values.
Instead of performing a single attention function with dmodel dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to dk , dk and dv dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding dv -dimensional output values. These are concatenated and once again projected, resulting in the final values.

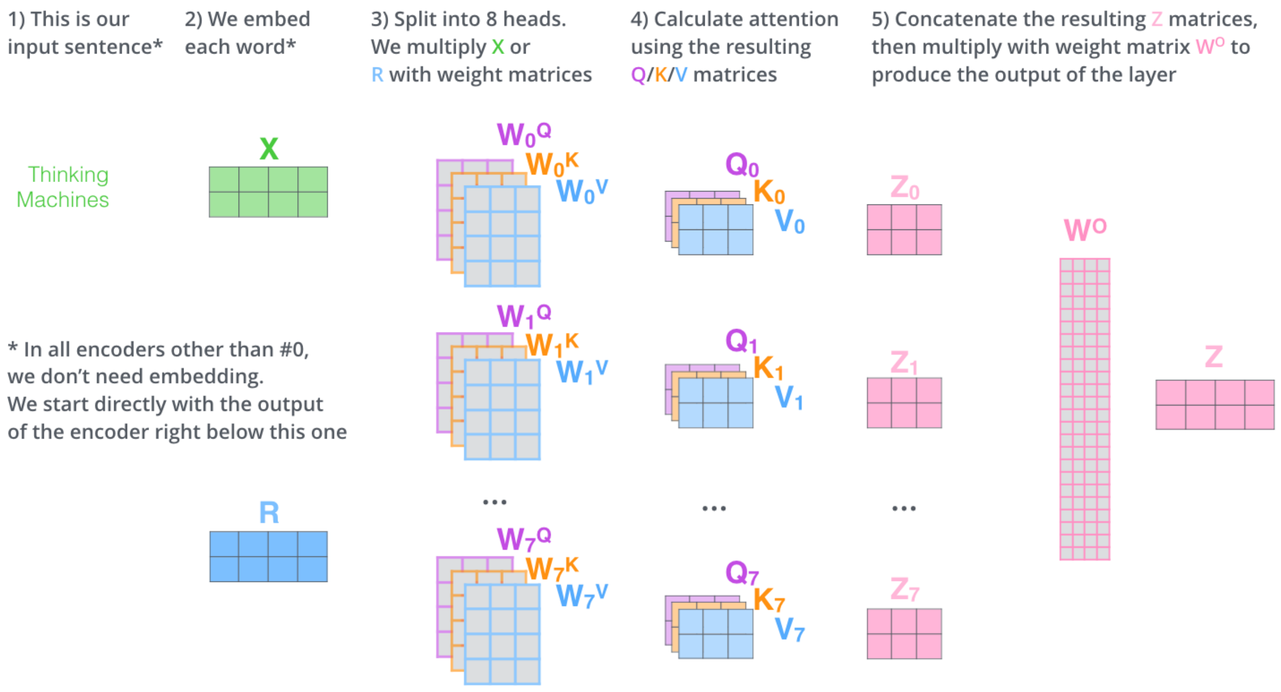

1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer
Thinking Machines
X
R
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
W0^Q W0^K W0^V
W1^Q W1^K W1^V
W7^Q W7^K W7^V
Q0 K0 V0
Q1 K1 V1
Q7 K7 V7
Z0
Z1
Z7
W^O
Z

Figure 4: Architecture of Multihead Attention vector calculations

3 Software Requirement Specification

Software Requirements Specification

for

Language Translation using Recurrent Neural Networks

Prepared by
Devashish Gaikwad: 111608023
Atharva Jadhav: 111608031

College Of Engineering, Pune

13th November, 2018
version 1.0

Table of Contents

1. Introduction

1.1 Purpose

The project aims to create a English to German and English to French bidirectional language translation model.

1.2 Document Conventions

Bold text has been used to emphasize section and subsection headings.
Highlights is to point out words in the glossary and italicized text is for used to
label and recognize diagrams.

1.3 Intended Audience and Reading Suggestions

This document is to be read by our project managers and project guides (teachers). The SRS has been organized
approximately in order of increasing specificity. The developers and project guides (teachers) need to become intimately familiar with the SRS.

1.4 Product Scope

The product - Translation Software will translate text between two languages, new accurate and faster algorithms for text translation will be used. The product will test and demonstrate the robustness and accuracy of the algorithms while providing the service via a webpage in a client-server system.

1.5 References

1. Cho, Merriënboer et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation” Cornell University Library, 2014
2. Manning, Luong et al. “Effective Approaches to Attention-based Neural Machine Translation” Cornell University Library, 2015
3. Sutskever, Vinyals et al. “Sequence to Sequence Learning with Neural Networks” Cornell University Library, 2015
4. Son, Allauzen et al. “Continuous Space Translation Models with Neural Networks”, Cornell University Library, 2012
5. Papineni, Ward, Zhu “BLEU: a Method for Automatic Evaluation of Machine Translation”, Cornell University Library, 2002

2. Overall Description

2.1 Product Perspective

The product uses Machine Translation where the same model can be used for translating between different language pairs. Models using recurrent neural networks are more accurate than conventional expert systems and feed forward neural network models. Client-server model would allow easy use of model and addition of more functionalities if needed.

2.2 Product Functions

1. English to German bidirectional text translation
2. English to French bidirectional text translation
3. Support for translating raw text
4. Simple server and website frontend

2.3 User Classes and Characteristics

This product can be used by anyone who wants to translate, learn French or German while knowing english. It can also be used for reverse translation. The user interface will be quite intuitive, so any advanced knowledge will not be necessary.

2.4 Operating Environment

The product will be designed for a unix based operating system. After a successful training and testing of model, a web server will be created to handle translation requests in text format will be created and hosted.

Specifications:

- For client / web page - any browser on any platform (mobile, tablet or PC) with input capabilities
- For Server and training the model - Environment with support for NVIDIA CUDA , Python version 3, Deep learning libraries for python, Flask micro web framework library for Python, Networking capabilities and internet connectivity

2.5 Design and Implementation Constraints

- Model training time - it can vary from hardware to hardware. Hardware not having NVIDIA GPU may take upto 10 times more time to train than hardware with NVIDIA GPU
- Therefore a Good GPU is necessary
- Language for the server will be python for the client languages will be in html, css, javascript
- Communication between client and server will be over HTTP using REST API

2.6 User Documentation

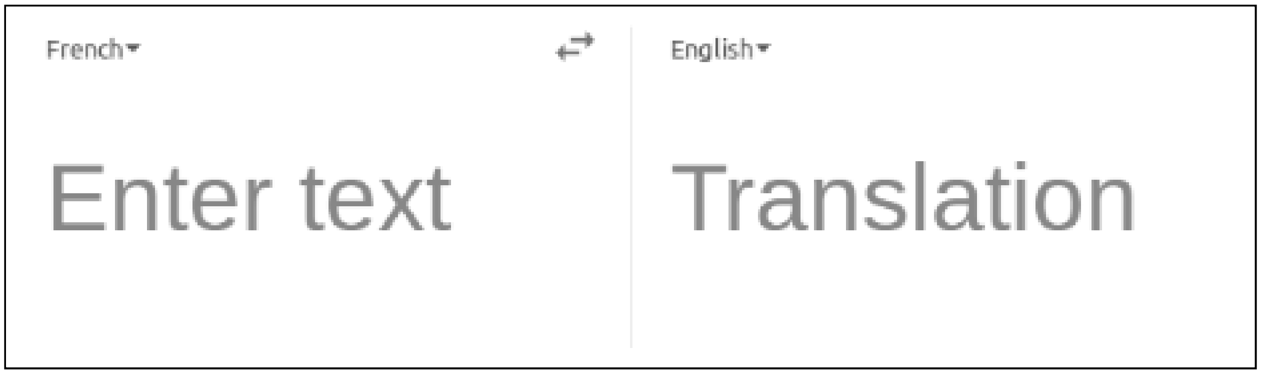
This product will include a help page.

2.7 Assumptions and Dependencies

- The server will run on a 64 bit Linux/Windows OS with support for Python Version 3 and support for NVIDIA CUDA and Deep learning libraries like Tensorflow, Theano, Cafe, etc
- Python libraries such as SimpleHTTPServer, Flask micro web framework, Threading Support along with Database connectivity and Deep learning libraries as Tensorflow, Theano, Cafe, etc
- AWS will be used for hosting website and server using EC2, S3 and Route 53 services of AWS
- We plan to use NVIDIA lab in college or AWS paid instance with a GPU to train our model.
- some basic components of model will be used from Deep learning libraries
- Bilingual parallel corpora datasets for training the model from websites like opus.nlpl.eu, nlp.stanford.edu, Official transcripts of European union parliament proceedings, etc

3. External Interface Requirements

3.1 User Interfaces



The user interface screen is described below:
1. Two interactive boxes to input in a language and obtain an output in other language.
2. Responsive Switch button to reverse the language translation either way.
3. Open help button for the user to help translate and add to the database.
4. Reactive Web app for satisfactory user interaction on any device which supports web browsing.

3.2 Hardware Interfaces

- supported device types -
  - client : all devices with a web browser
  - server : windows/linux servers which satisfy hardware requirements
- nature of data -
  - client - for communication between client server the data will be text and web api requests
  - server - data in training dataset is text(strings), for communication with client web api will be used
- control interactions between the software and the hardware -
  - client - Use of touchscreen or Keyboard to navigate the web page e.g enter the text and submit the translation request.
  - server - NVIDIA CUDA to exploit the capabilities of the graphics card, terminal for controlling the server
- Communication protocols - HTTP protocol for serving REST API will be used for communication between client and server.

3.3 Software Interfaces

- Databases -
  - server - text, csv files of bilingual parallel corpora will be used for training
- OS -
  - client - any operating system which supports web browser
  - server - Windows / Ubuntu / Debian 64 bit
- Tools -
  - Jupyter notebook - for visualizing the model
  - Webflow, Mockflow, Mockplus - for designing the UI
  - AWS console - for monitoring the AWS instances
  - Terraform - managing AWS instances
- libraries -
  - server -
    - Tensorflow
    - Theano
    - Cafe
    - Scikit - Learn
    - Flask
    - pandas
    - SimpleHTTPServer
- Commercial Components -
  - AWS Cloud services for hosting website and server
    - respective costs for each instance and hosting
- API requests -
  - to server -
    - translate - includes text to be translated and language pair
  - to client -
    - answer - translated text

3.4 Communications Interfaces

- Standard -
  - between client and server - HTTP
- Requirement -
  - client - have a web browser
  - server - have a web server capable of handling requests and replying back with data
- Webpage on the client device with a form like interface
- Encryption - TBD as ssl certificate is required for HTTPS

4. System Features

4.1 System Features

REQ#	Feature	Priority
1	Correct translation from one language to another	High
2	Switch between different languages	High
3	Create and train model as per reference papers	High
4	Server for connecting frontend to model	High
5	Reactive website to act as frontend	Medium
6	Download original text and translated text as a text file	Medium
7	User submitted translations to help train the model	Medium
8	Parallelization for serving multiple requests simultaneously	Low
9	Feedback implementation	Low

4.2 Stimulus/Response Sequences -

1. Successful Translation – Will lead to the immediate translated output.
2. Switching Languages - Drop down will appear giving us option about other languages.
3. Button - clicking on reverse button will transverse languages.
4. Download Button - clicking will Download original text and translated text as a text file
5. Help us Button - clicking will open another page/dialog box via which user will send us translated text
6. Feedback button - clicking will open dialog box for giving feedback

4.3 Functional Requirements for *The User*:

1. have a device with web browser
2. access to internet
3. access to website of this translation service

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Server
  - Model training time - it can vary from hardware to hardware. Hardware not having NVIDIA GPU may take upto 10 times more time to train than hardware with NVIDIA GPU
  - Thus a Good GPU is necessary
  - Language for the server will be python for the client languages will be in html, css, javascript
  - Communication between client and server will be over HTTP using REST API
  - The server will run on a 64 bit Linux/Windows OS with support for Python Version 3 and support for NVIDIA CUDA and Deep learning libraries like Tensorflow, Theano, Cafe, etc
  - Python libraries such as SimpleHTTPServer, Flask micro web framework, Threading Support along with Database connectivity and Deep learning libraries as Tensorflow, Theano, Cafe, etc
  - minimum instance requirements for server :
    - RAM : 8GB
    - Processor Cores : 4
    - Storage : 250GB
    - GPU : NVIDIA CUDA capable or OpenCL capable
    - Networking : High bandwidth - 1 Gbps
  - ideal AWS instances : p2.xlarge, g3s.xlarge
  - GPU models : NVIDIA Tesla v100, P4, P100, K80
- Client:
  - any device having a web browser

5.2 Security Requirements

Access to text translation database will only be given to developers, there is no provision for user login and all the previous searches will be stored in cookies, thus giving user the freedom to control their own search history.

5.3 Software Quality Attributes

Interface usability, BLEU score of translation model, reliability, webpage portability between different devices, availability, correctness, flexibility, interoperability, maintainability

5.4 Business Rules

The website will available to the everyone 24/7 irrespective of their device. They can translate and download text from anywhere with active internet connection. The software will be completely feasible for everyone because of its need and necessity. The application will be Open source and under GNU GPL v3

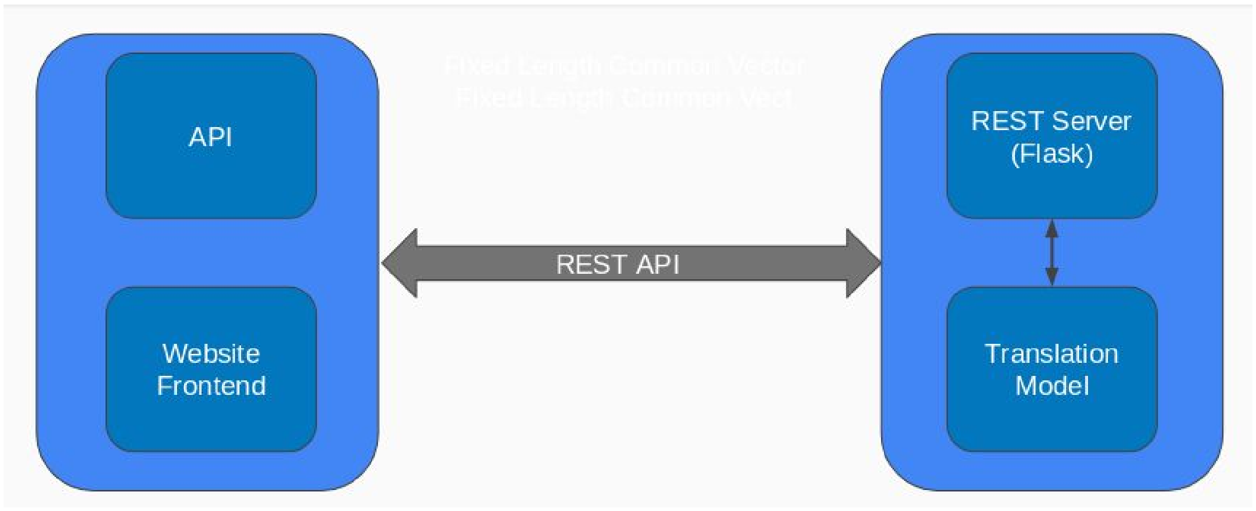
6. Other Requirements

No other requirements.

Appendix A: Glossary

All common software acronyms apply

Appendix B: Analysis Models



Appendix C: To Be Determined List

1. Cloud instance provider - As default we will use AWS but if Google cloud is able to provide cloud instances with cheaper cost we will use that.
2. Windows/Linux for server - will be decided on the instance costs and performance of Deep learning libraries

4 Literature Survery

Literature Survey

Research Paper 1: Speechalator: two-way speech-to-speech translation on a consumer PDA

Alex Waibel, Ahmed Badran et al. Carnegie Mellon University, Pittsburgh

This paper describes a working two-way speech-to-speech translation system that runs in near real-time on a consumer handheld computer. It can translate from English to Arabic and Arabic to English in the domain of medical interviews. It also describes the general architecture and frameworks within which we developed each of the components: HMM-based recognition, interlingua translation (both rule and statistically based), and unit selection synthesis.

Research Paper 2: Exploiting Similarities among Languages for Machine Translation

Tomas Mikolov, Quoc V. Le, Ilya Sutskever: Google Inc., Mountain View

This paper develops a method that can automate the process of generating and extending dictionaries and phrase tables. Our method can translate missing word and phrase entries by learning language structures based on large monolingual data and mapping between languages from small bilingual data. It uses distributed representation of words
and learns a linear mapping between vector spaces of languages.

Research Paper 3: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre,Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio

The paper proposes a novel neural network model called RNN Encoder−Decoder that consists of two recurrent neural networks (RNN).

One RNN encodes a sequence of symbols into a fixed length vector representation, and the other decodes the representation into another sequence of symbols. The encoder and decoder of the proposed model are jointly trained to maximize the conditional probability of a target sequence given a source sequence.

Research Paper 4: Neural Machine Translation by jointly learning to align and translate.

Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio

In this paper, authors conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly.

Research Paper 5: Effective Approaches to Attention-based Neural Machine Translation

Minh-Thang Luong, Hieu Pham, Christopher D. Manning

An attentional mechanism has lately been used to improve neural machine translation (NMT) by selectively focusing on parts of the source sentence during translation. However, there has been little work exploring useful architectures for attention-based NMT. This paper examines two simple and effective classes of attentional mechanism: a global approach which always attends to all source words and alocalone that only looks at a subset of source words at a time. With local attention, we achieve a significant gain of 5.0 BLEU points over non-attentional systems that already incorporate known techniques such as dropout. Our ensemble model using different attention architectures yields a new state-of-the-art result inthe WMT’15 English to German translation task with 25.9 BLEU points, an improvement of 1.0 BLEU points over the existing best system backed by NMT andann-gram reranker.

Research Paper 6 : Sequence to Sequence Learning With Neural Networks

Ilya Sutskever, Oriol Vinyals, Quoc V. Le

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, Authors present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Authors' method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Main result is that on an English to French Translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set.

Research Paper 7 : Continuous Space Translation Models with Neural Networks

Le Hai Son, Alexandre Allauzen, François Yvon

The use of conventional maximum likelihood estimates hinders the performance of existing phrase-based translation models. For lack of sufficient training data, most models only consider a small amount of context. As a partial remedy, we explore here several continuous space translation models, where translation probabilities are estimated using a continuous representation of translation units in lieu of standard discrete representations

Research Paper 8 : Attention Is All You Need

Ashish Vaswani, Jakob Uszkoreit, Aidan N. Gomez, et al

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English- to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Research Paper 9 : Scaling Neural Machine Translation

Myle Ott, Sergey Edunov, et al

Sequence to sequence learning models still require several days to reach state of the art performance on large benchmark datasets using a single machine. This paper shows that reduced precision and large batch training can speedup training by nearly 5x on a single 8- GPU machine with careful tuning and implementation On WMT'14 English-German translation, we match the accuracy of Vaswani et al. (2017) in under 5 hours when training on 8 GPUs and we obtain a new state of the art of 29.3 BLEU after training for 85 minutes on 128 GPUs. We further improve these results to 29.8 BLEU by training on the much larger Paracrawl dataset. On the WMT'14 English- French task, we obtain a state-of-the-art BLEU of 43.2 in 8.5 hours on 128 GPUs.

Research paper 10 : BLEU: a Method for Automatic Evaluation of Machine Translation

Kishore Papineni,Salim Roukos,Todd Ward, and Wei-JingZhu

The BLEU metric ranges from 0 to 1 (generally multiplied by 100). Few translation will attain a score of 1 unless they are identical to a reference translation. For this reason,even a human translator will not necessarily score 1. It is important to note that the more reference translations persentence there are, the higher the score is.

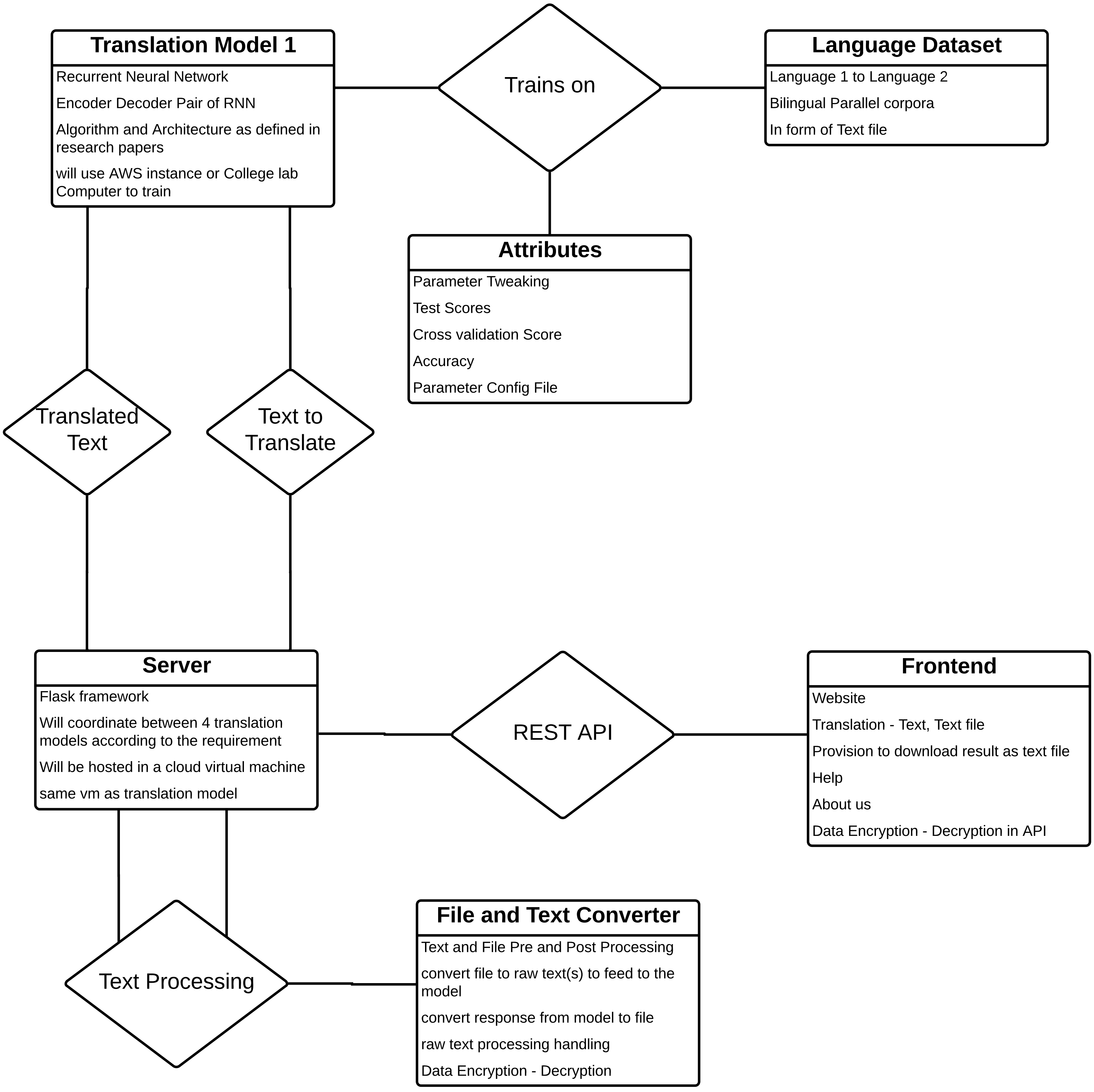
Survey:

	Speechalator: two way speech to speech translation on a consumer PDA	Exploiting similarities among Languages for Machine Translation	Neural Machine Translation by jointly learning to align and translate.	Effective Approaches to Attention based Neural Machine Translation	Sequence to Sequence Learning With Neural Networks	Continuous Space Translation Models with Neural Networks	Learning Phrase Representations using RNN Encoder Decoder
Translation with context	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Bidirectional Translation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Model Domain	No (Rule Based)	Dictionary Based (Rule based)	RNN (Deep Learning)	RNN + LSTM + Feedback (Deep Learning)	RNN + LSTM (Deep Learning)	ANN + Vectorization (Machine Learning)	RNN + LSTM + Feedback (Deep Learning)
Encoder Decoder	No(Rule Based)	No	Yes	Yes	Yes	No (SOUL model)	Yes
Multiple Language Pairs	No	No	Yes	Yes	Yes	Yes	Yes
Accuracy	Not Available	Not Available	32.42	25.9	30.6	26.8	34.54
Primary Language	English to Arabic	English to Spanish	English to French	English to German	English to French	English to French	English to French
Voice based	Yes	No	No	No	No	No	No
Hardware	Old PDAs (ARM machines)	Old CPUs - low power req	Only CPU	GPU	GPU	GPU	GPU
Paper Year	Sept 2003	Sep 2013	Oct 2014	Sept 2015	Dec 2014	June 2012	May 2016

5 Architectural Diagram

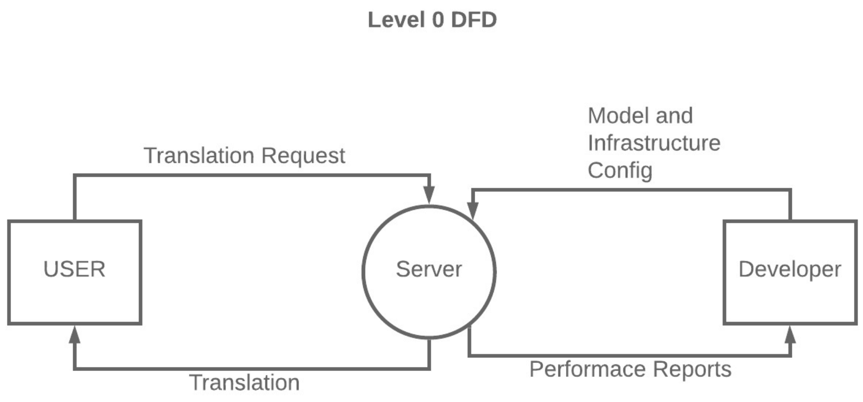
Architecture ER Diagram
Language Translation using Recurrent Neural Networks

Devashish Gaikwad : 111608023
Atharva Jadhav : 111608031

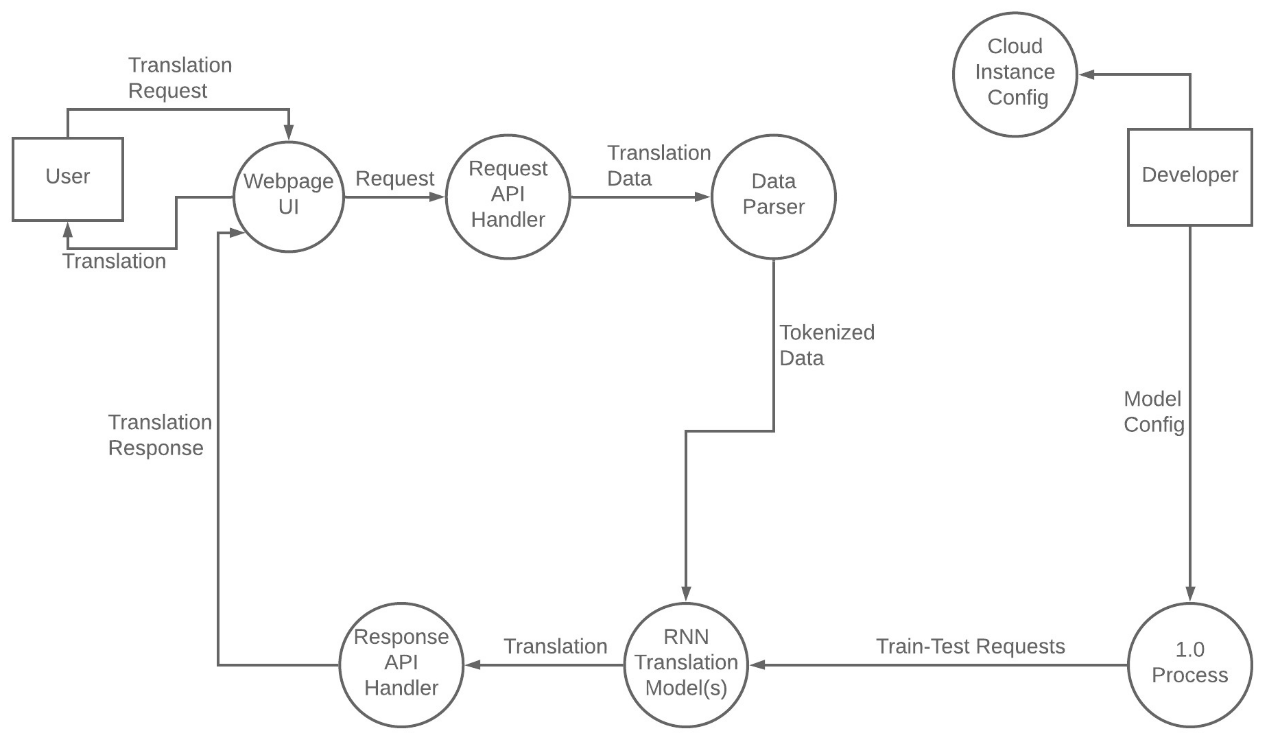


Translation Model 1
Recurrent Neural Network
Encoder Decoder Pair of RNN
Algorithm and Architecture as defined in research papers
will use AWS instance or College lab Computer to train
Trains on
Language Dataset
Language 1 to Language 2
Bilingual Parallel corpora
In form of Text file
Attributes
Parameter Tweaking
Test Scores
Cross validation Score
Accuracy
Parameter Config File
Translated Text
Text to Translate
Server
Flask framework
Will coordinate between 4 translation models according to the requirement
Will be hosted in a cloud virtual machine
same vm as translation model
REST API
Frontend
Website
Translation - Text, Text file
Provision to download result as text file
Help
About us
Data Encryption - Decryption in API
Text Processing
File and Text Converter
Text and File Pre and Post Processing
convert file to raw text(s) to feed to the model
convert response from model to file
raw text processing handling
Data Encryption - Decryption

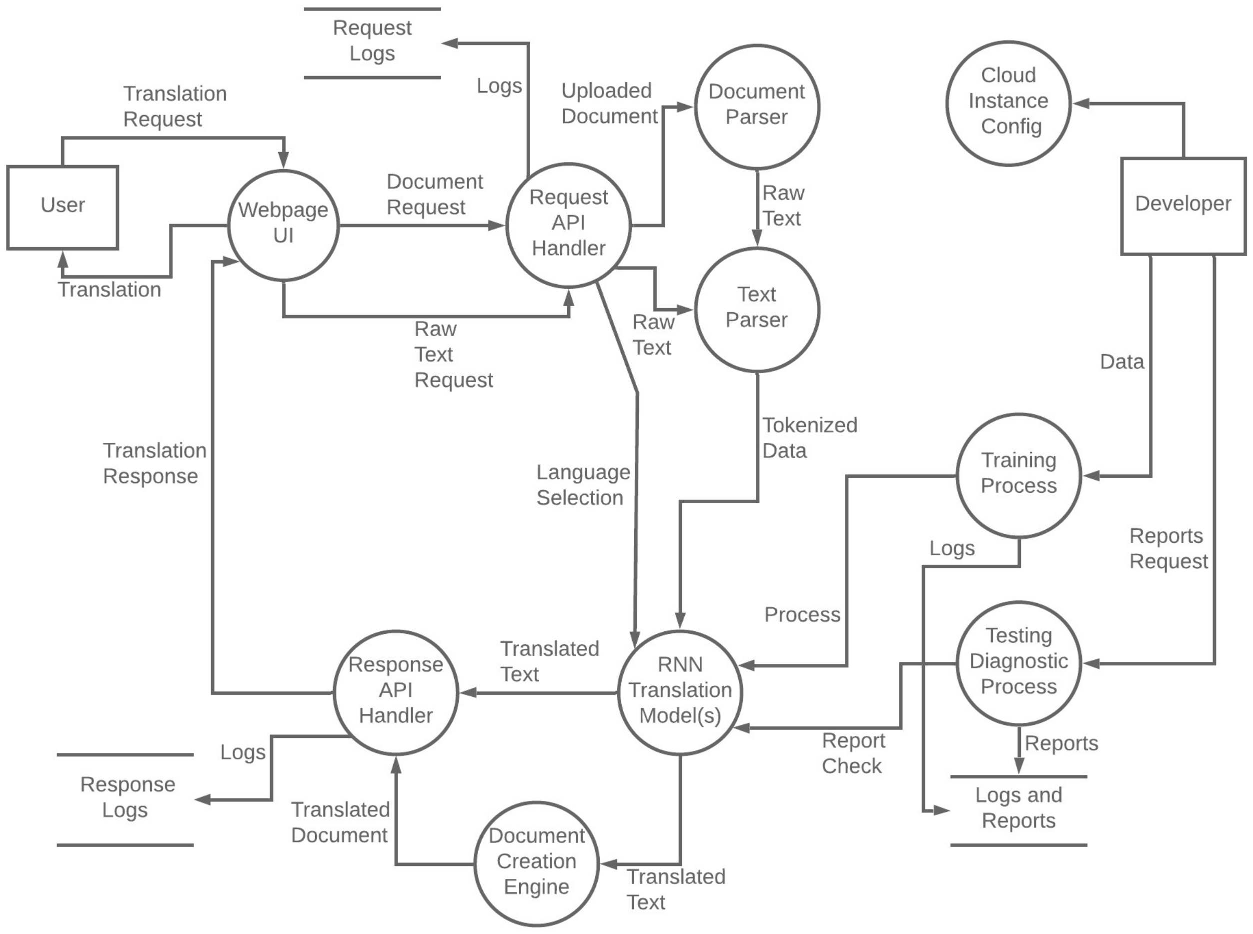
6 Data Flow Diagram



Level 1 DFD

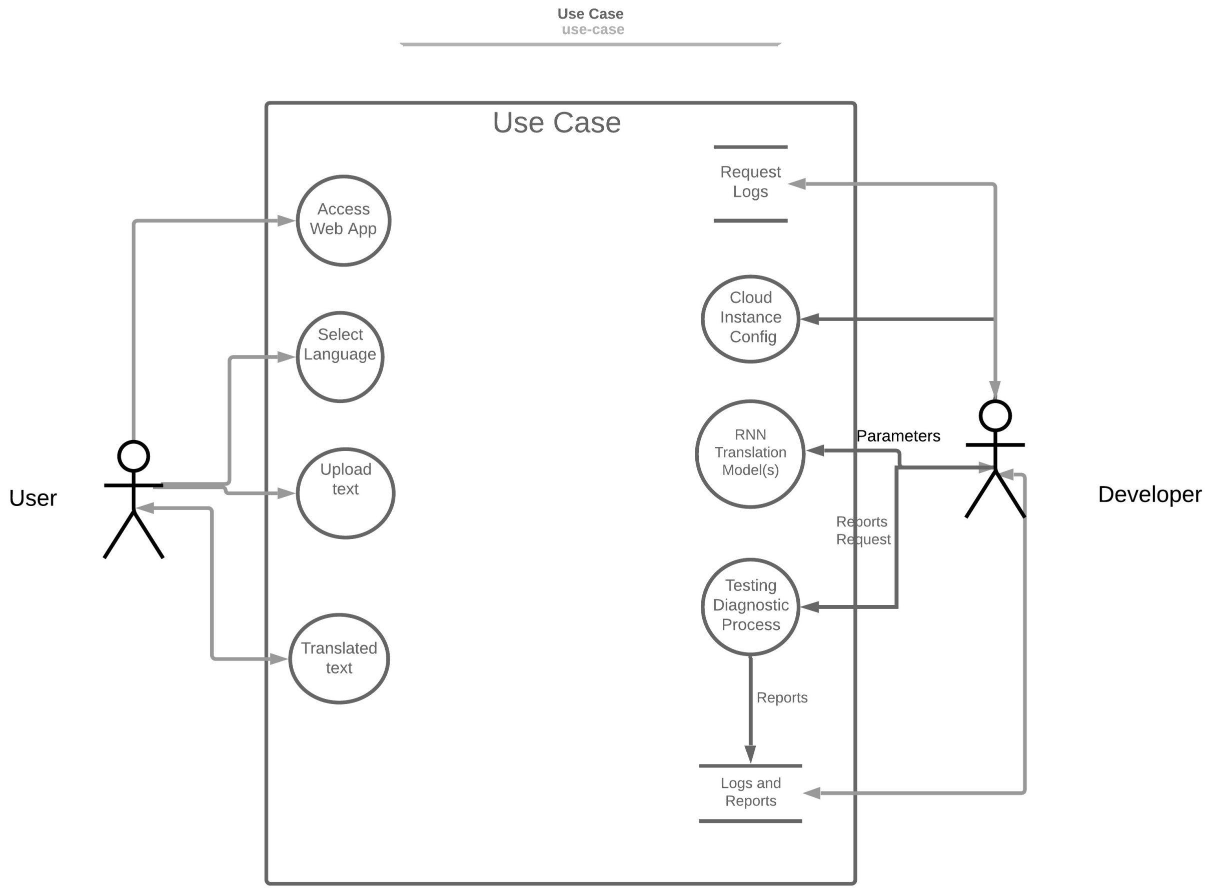

User
Translation Request
Translation
Webpage UI
Request
Request API Handler
Translation Data
Data Parser
Tokenized Data
Cloud Instance Config
Developer
Model Config
Translation Response
Response API Handler
Translation
RNN Translation Model(s)
Train-Test Requests
1.0 Process

Level 2 DFD

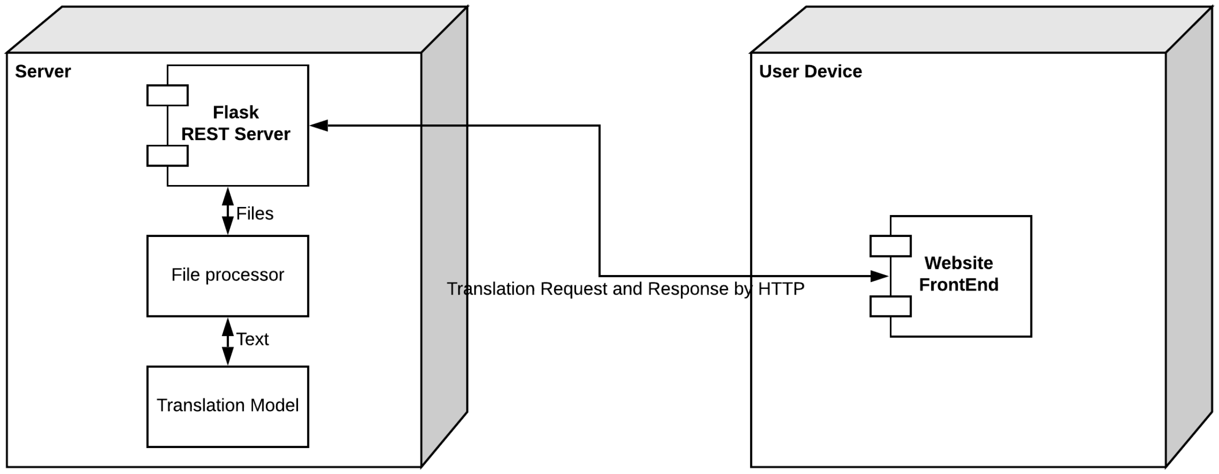

Request Logs
Logs
Uploaded Document
Document Parser
Cloud Instance Config
Translation Request
User
Webpage UI
Document Request
Request API Handler
Raw Text
Developer
Translation
Text Parser
Raw Text
Raw Text Request
Data
Tokenized Data
Translation Response
Language Selection
Training Process
Logs
Reports Request
Process
Testing Diagnostic Process
Translated Text
Response API Handler
RNN Translation Model(s)
Logs
Report Check
Reports
Response Logs
Translated Document
Logs and Reports
Document Creation Engine
Translated Text

7 UML Diagrams

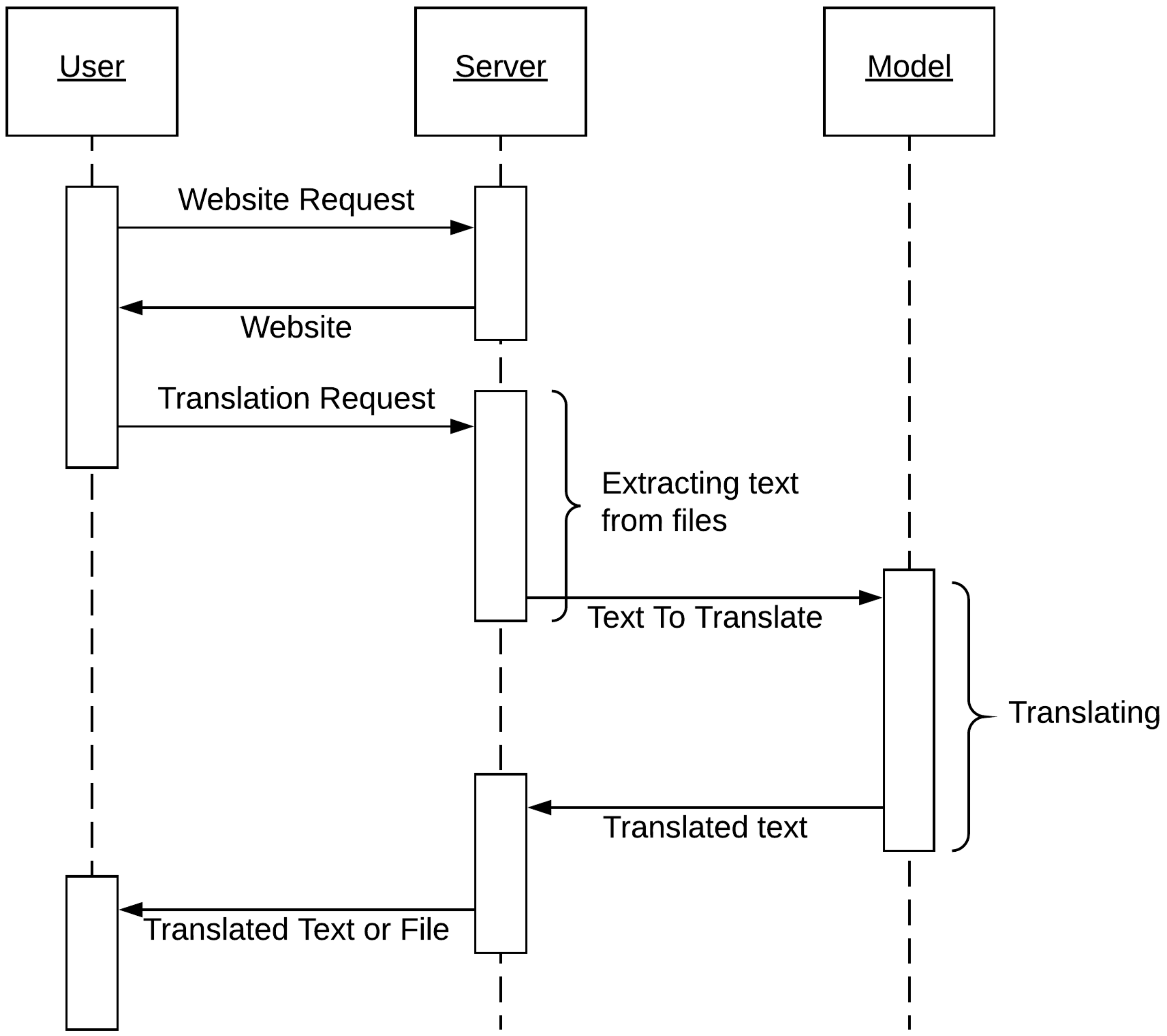
7.1 Use Case Diagram



7.2 Deployment Diagram

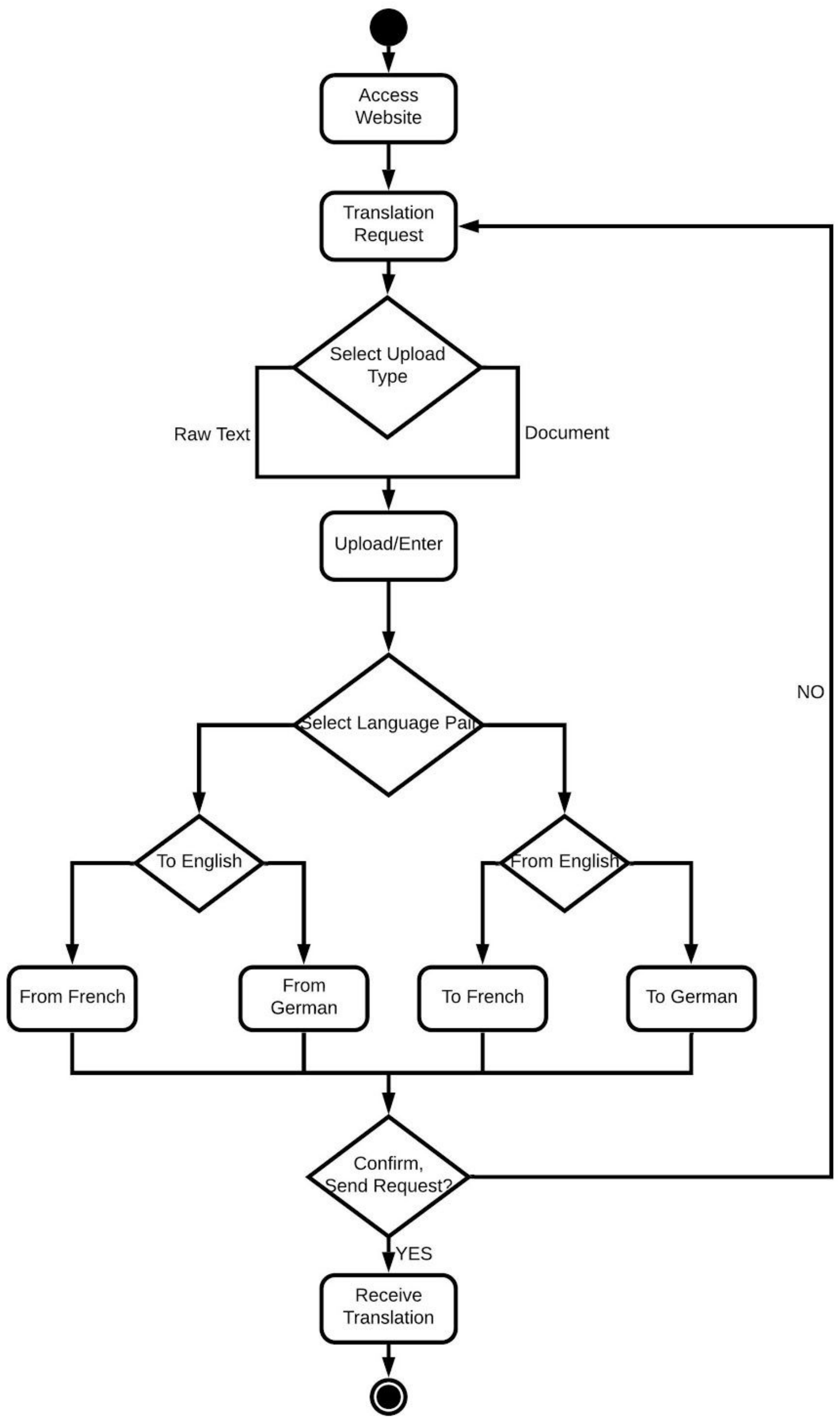


7.3 Sequence Diagram

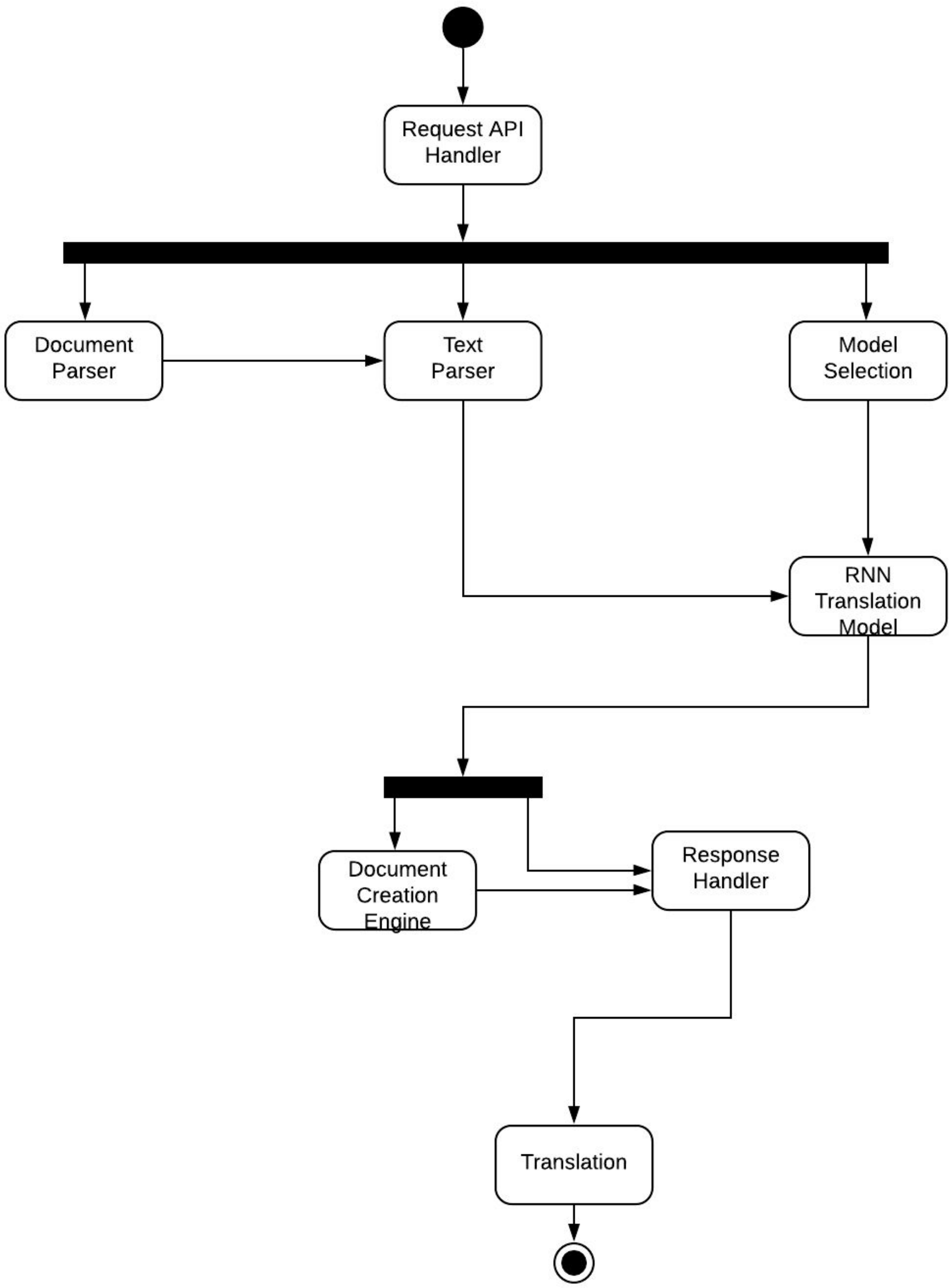


7.4 Activity Diagram

USER ACTIVITY DIAGRAM

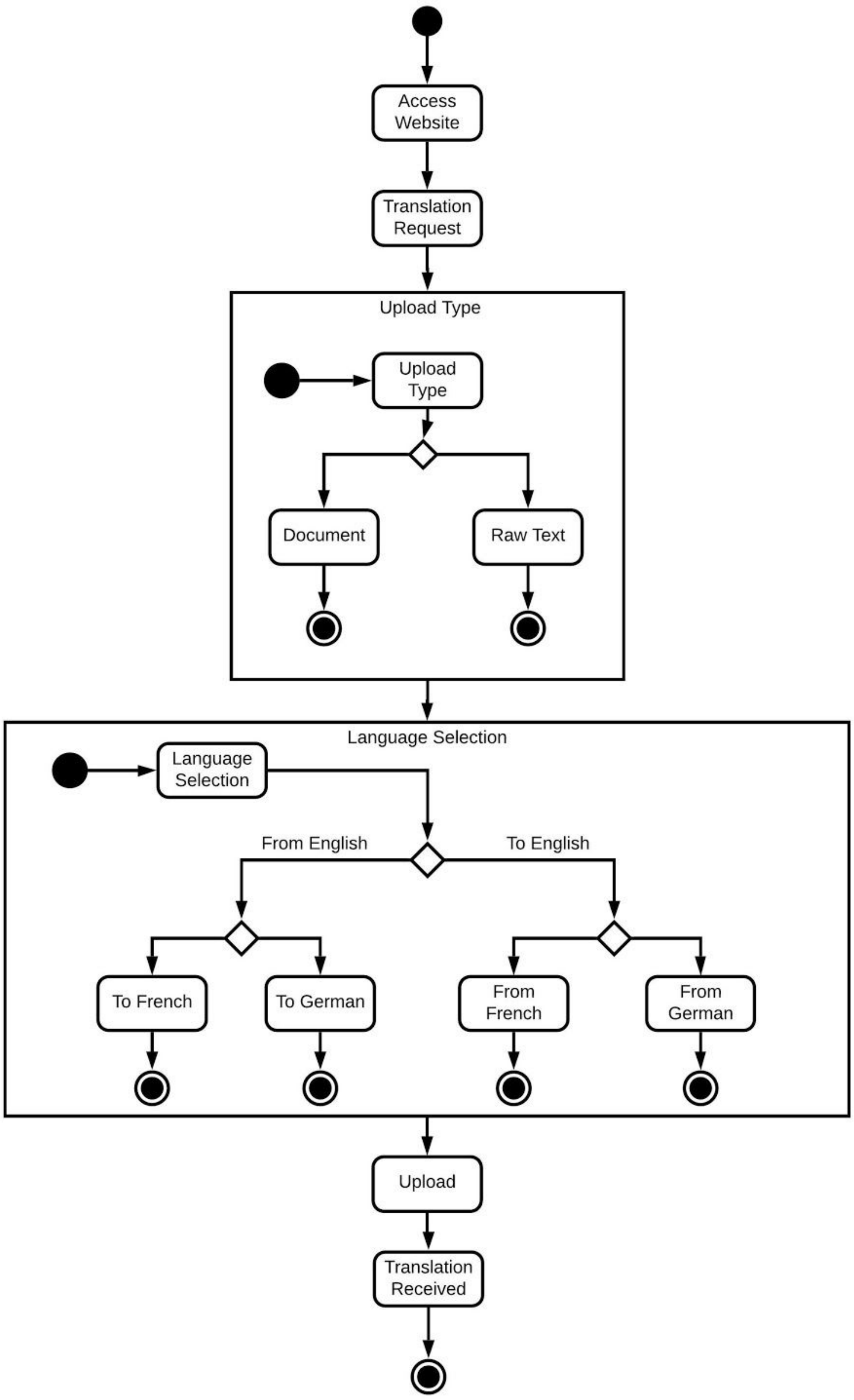


Developer Activity Diagram

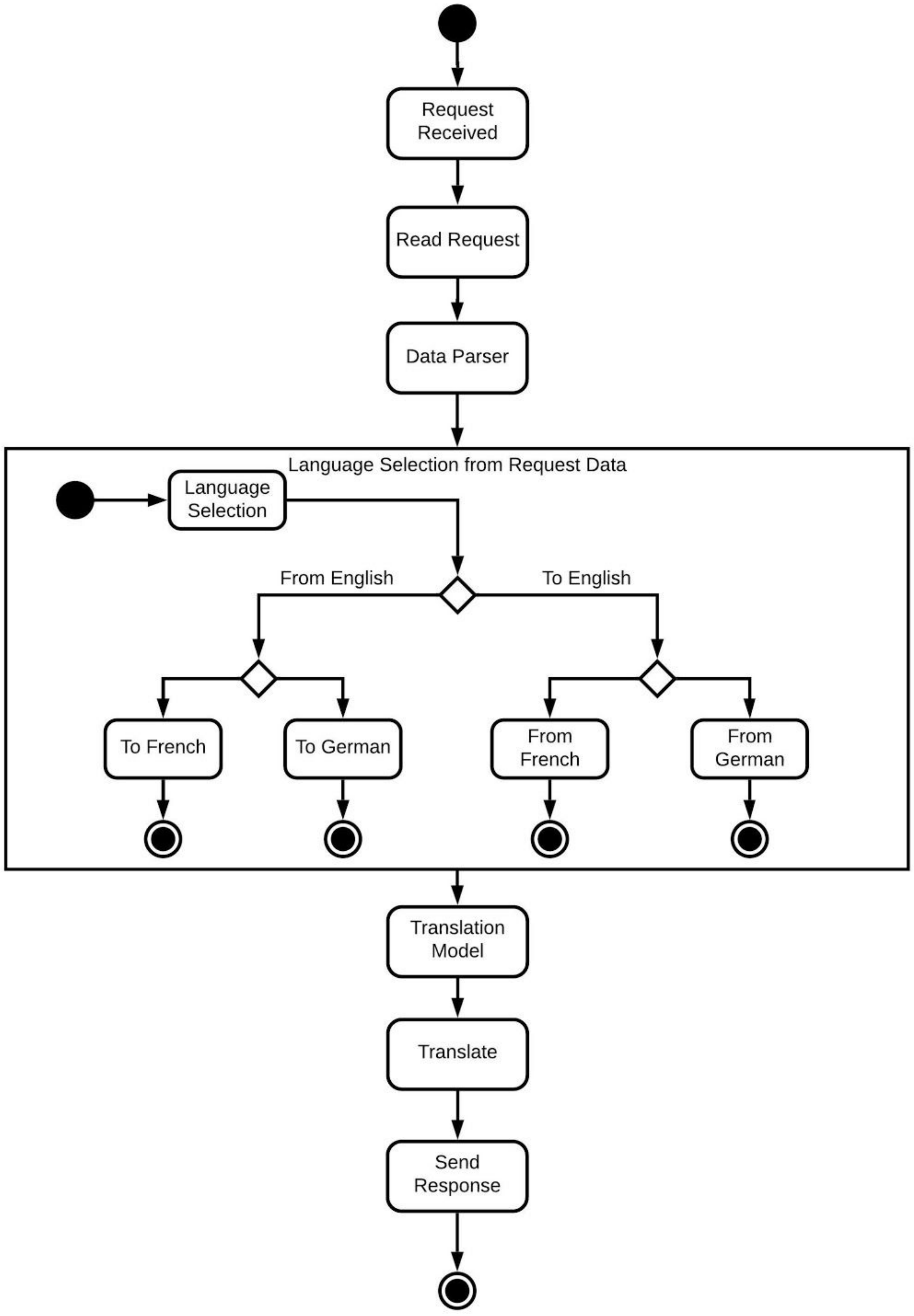


7.5 State Diagram

State Diagram for User

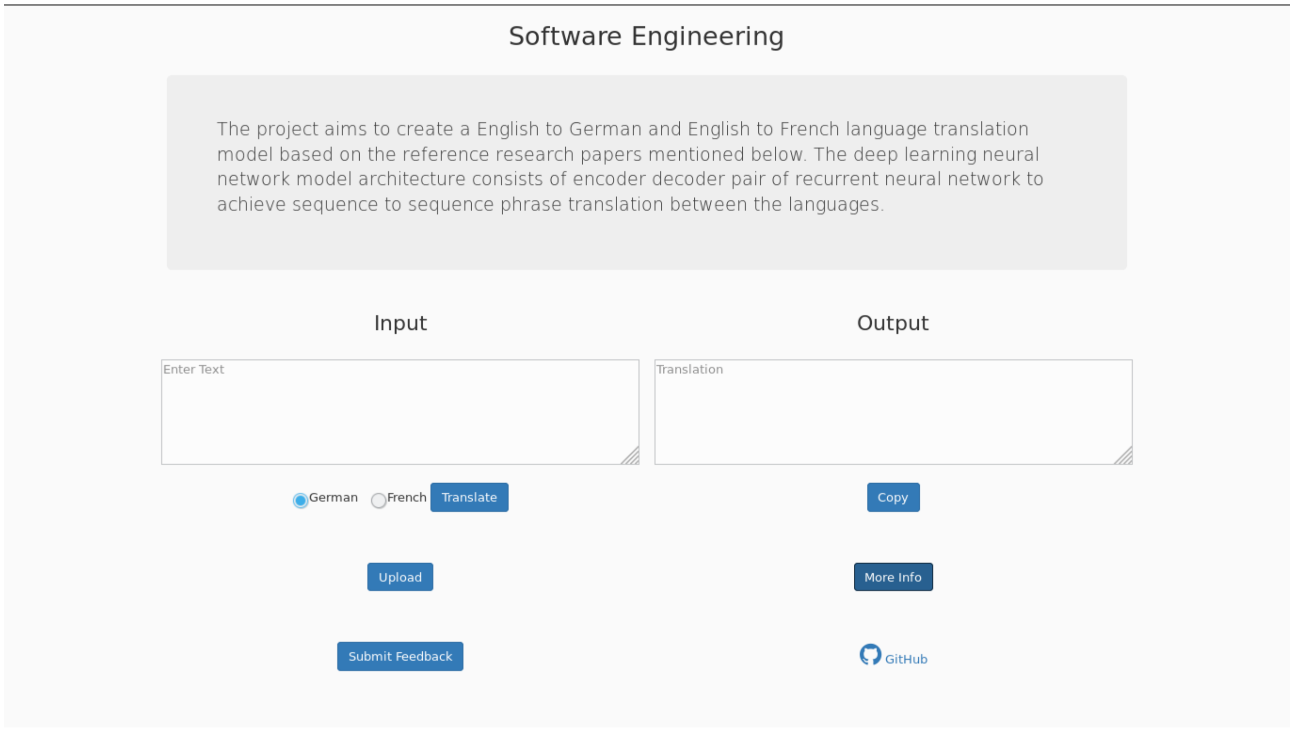


State Diagram for Server



8 Implementation Screenshots

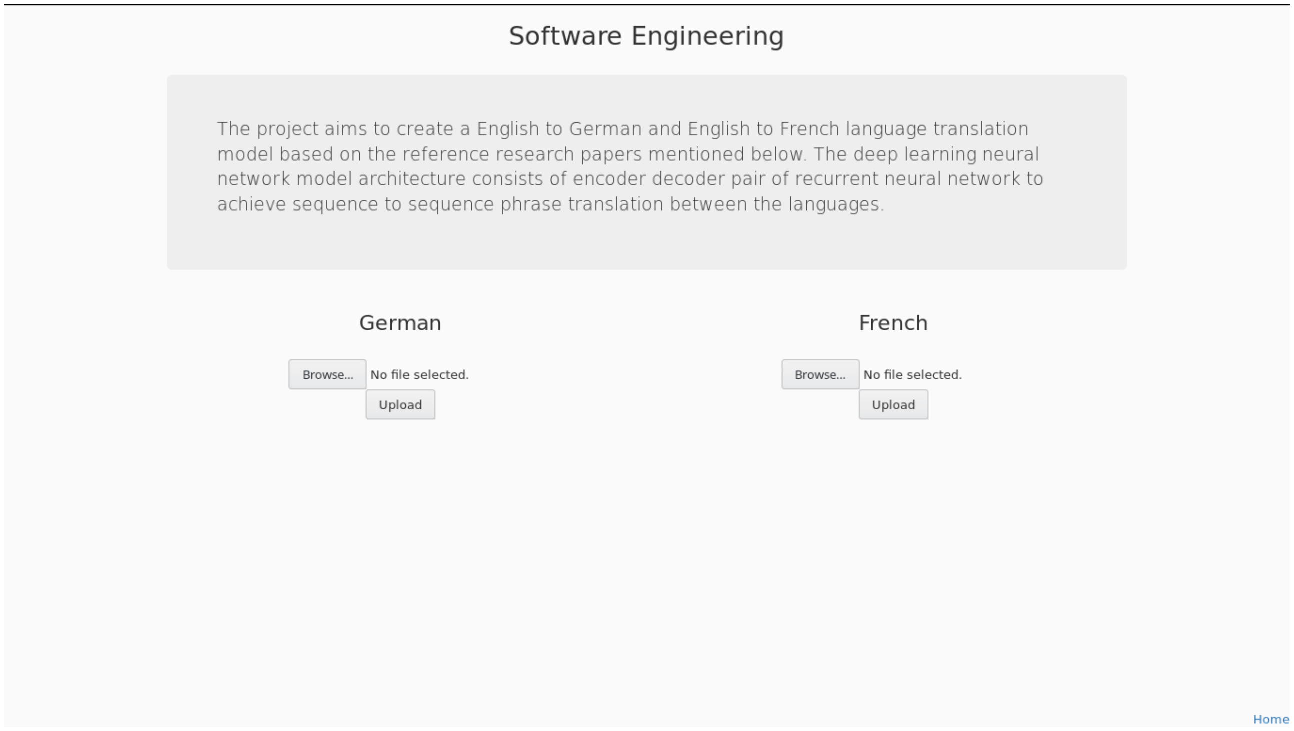
8.1 Index Page



Software Engineering

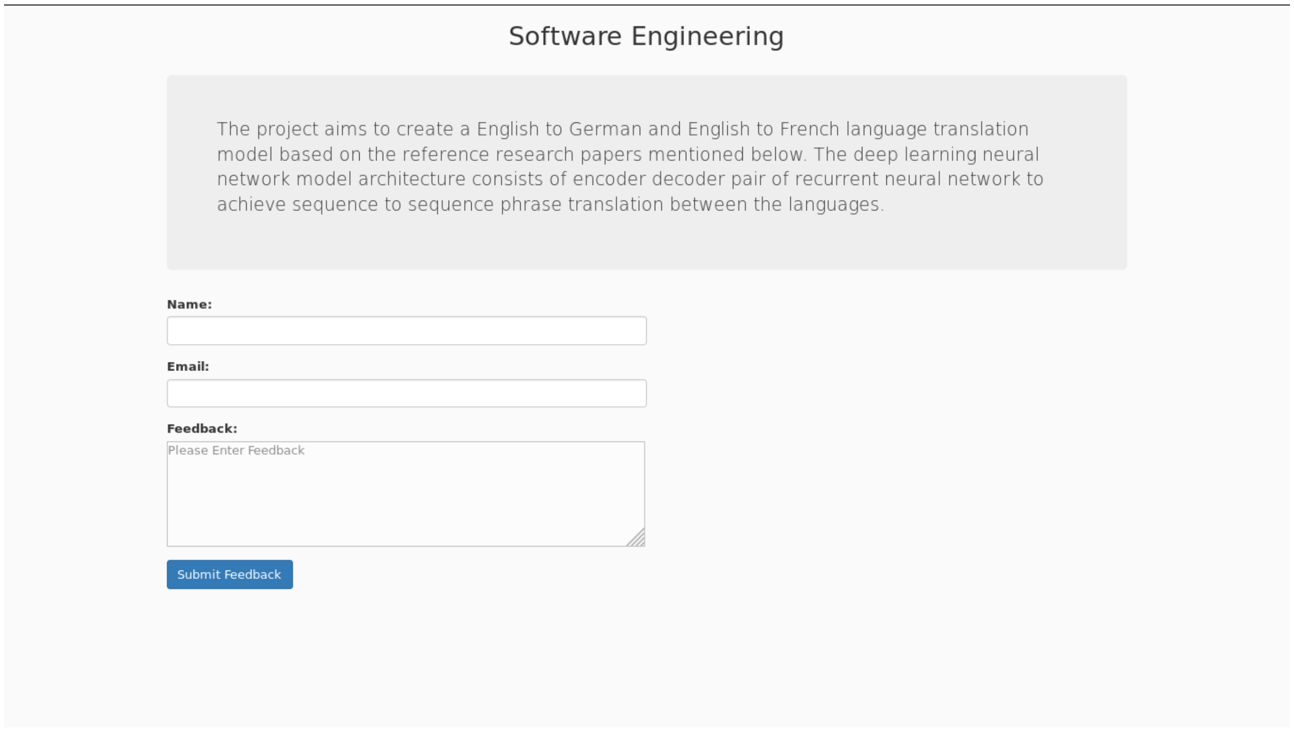
The project aims to create a English to German and English to French language translation model based on the reference research papers mentioned below. The deep learning neural network model architecture consists of encoder decoder pair of recurrent neural network to achieve sequence to sequence phrase translation between the languages.

8.2 Upload Page


Software Engineering

The project aims to create a English to German and English to French language translation model based on the reference research papers mentioned below. The deep learning neural network model architecture consists of encoder decoder pair of recurrent neural network to achieve sequence to sequence phrase translation between the languages.

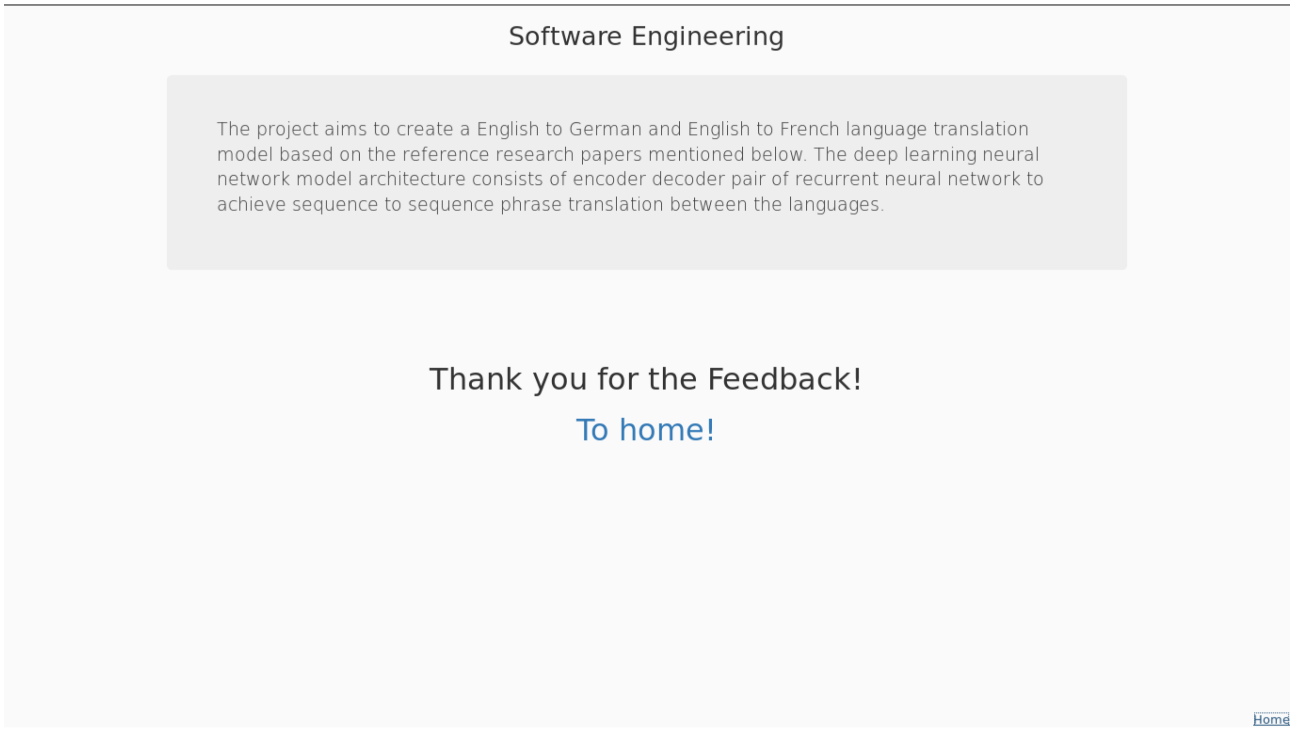
8.3 Feedback Page



Software Engineering

The project aims to create a English to German and English to French language translation model based on the reference research papers mentioned below. The deep learning neural network model architecture consists of encoder decoder pair of recurrent neural network to achieve sequence to sequence phrase translation between the languages.

8.4 Thank You Page

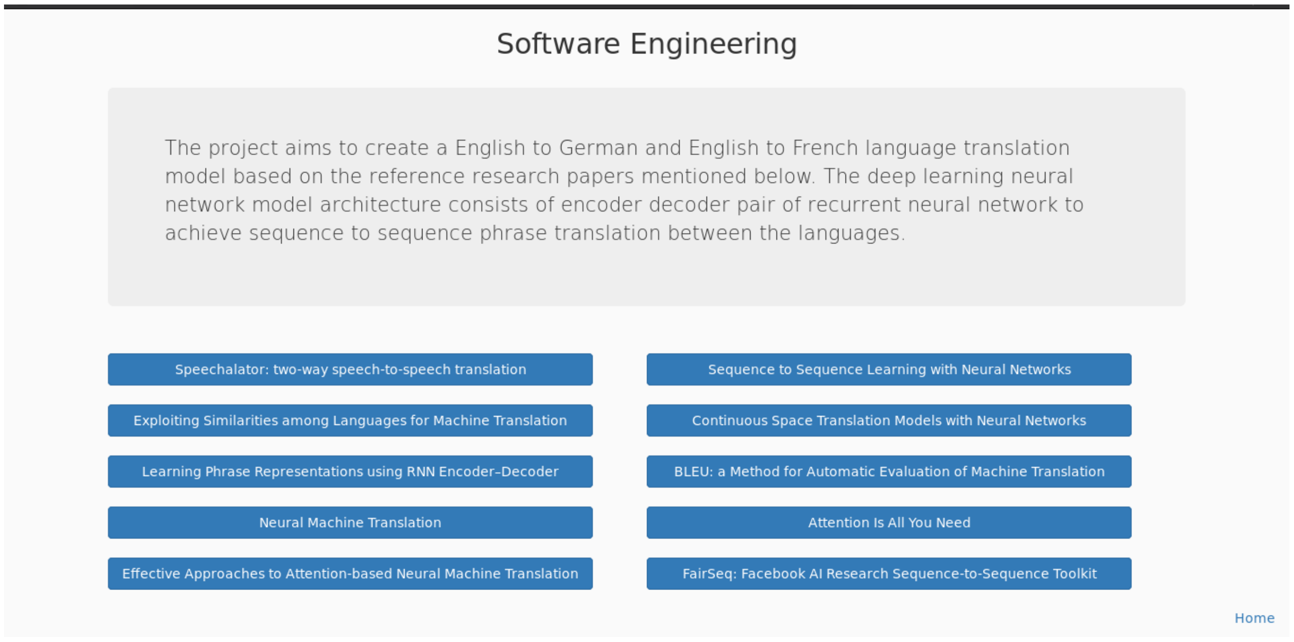

Software Engineering

The project aims to create a English to German and English to French language translation model based on the reference research papers mentioned below. The deep learning neural network model architecture consists of encoder decoder pair of recurrent neural network to achieve sequence to sequence phrase translation between the languages.

Thank you for the Feedback!

To home!

8.5 References Page


Software Engineering

The project aims to create a English to German and English to French language translation model based on the reference research papers mentioned below. The deep learning neural network model architecture consists of encoder decoder pair of recurrent neural network to achieve sequence to sequence phrase translation between the languages.

Speechalator: two-way speech-to-speech translation
Sequence to Sequence Learning with Neural Networks
Exploiting Similarities among Languages for Machine Translation
Continuous Space Translation Models with Neural Networks
Learning Phrase Representations using RNN Encoder–Decoder
BLEU: a Method for Automatic Evaluation of Machine Translation
Neural Machine Translation
Attention Is All You Need
Effective Approaches to Attention-based Neural Machine Translation
FairSeq: Facebook AI Research Sequence-to-Sequence Toolkit
Home

9 Project Estimation

9.1 Lines of Code

The project contains various files in python and HTML.

Language	**Files**	**Blank**	**Comment**	**Code**
HTML	6	24	0	419
Python	4	103	56	395
Sum	10	127	56	814

9.2 COCOMO Model

The basic COCOMO model aims at estimating software development, in a quick fashion.
The basic COCOMO equations take the form:
E = $a_b(LOC)^b b$
D = $c_b(E)^d b$
S = E / D people
P = LOC/E

Project	**a**	**b**	**c**	**d**
Organic Mode	2.4	1.05	2.5	0.38
Semi Detached Mode	3.0	1.12	2.5	0.35
Embedded Mode	3.0	1.20	2.5	0.32

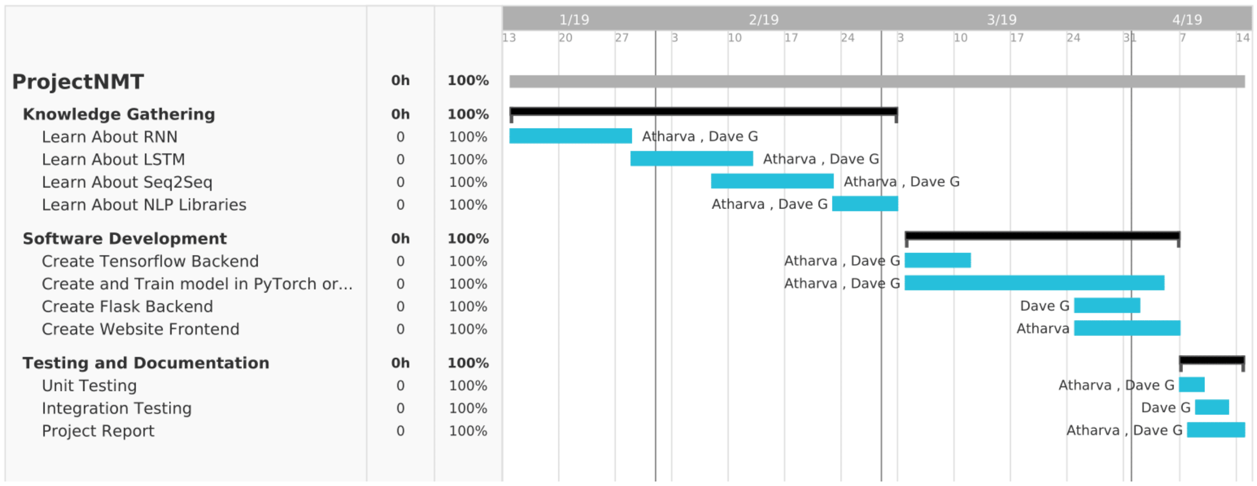
9.3 Function Point Analysis

The number of function points for the project considering complexity, feasibility, re-usability, performance and communication is 313.54

10 Project Scheduling

10.1 Gantt Chart

Here, the users are:
”Dave G” as Devashish Gaikwad
”Atharva” as Atharva Jadhav



ProjectNMT	**0h**	**100%**
Knowledge Gathering	**0h**	**100%**
Learn About RNN	0	100%
Learn About LSTM	0	100%
Learn About Seq2Seq	0	100%
Learn About NLP Libraries	0	100%
Software Development	**0h**	**100%**
Create Tensorflow Backend	0	100%
Create and Train model in PyTorch or...	0	100%
Create Flask Backend	0	100%
Create Website Frontend	0	100%
Testing and Documentation	**0h**	**100%**
Unit Testing	0	100%
Integration Testing	0	100%
Project Report	0	100%

11 Testing

11.1 Unit Tests

These are some of the unit tests conducted and passed :

Test ID	Test Case Input	Expected Output	Output
test_driver_post_apos	”&apos; ”	”’”	”’”
test_driver_post_atat	”@@”	””	””
test_driver_post_n	”\n”	” ”	” ”
test_driver_post_atdat	”@-@”	””	””
test_driver_trans_nos	”123456879”	” ”	” ”
test_driver_trans_nos_dash	”12345687–9”	” ”	” ”
test_allowed_fname_txt	”dev.txt”	”True”	”True”
test_allowed_fname_pdf	”dev.pdf”	”True”	”True”
test_allowed_fname_png	”dev.png”	”True”	”True”
test_allowed_fname_jpg	”dev.jpg”	”True”	”True”
test_allowed_fname_jpeg	”dev.jpeg”	”True”	”True”
test_txt_to_txt	”unit_test_trial.txt”	”This is a command line entry point.”	”This is a command line entry point.”
test_is_pdf	”dev.pdf”	”True”	”True”

11.2 Integration Tests

These are some of the integration tests conducted and passed :

Test ID	Test Case Input	Expected Output	Output
En_to_De 1	”Today is monday”	”Heute ist Montag”	”Heute ist Montag”
En_to_De2	”Today is a beautiful day”	”Heute ist ein schöner Tag”	”Heute ist ein schöner Tag”
En_to_Fr1	”Today is monday”	”Aujourd’hui c’est un lundi”	”Aujourd’hui c’est un lundi”
En_to_Fr2	”Today is a beautiful day”	”Aujourd’hui est un beau jour”	”Aujourd’hui est un beau jour”

12 Reference Papers

1. Alex Waibel, Ahmed Badran et al. “Speechalator: two-way speech-to-speech translation on a consumer PDA” Carnegie Mellon University, 2003

2. Tomas Mikolov, Quoc V. Le, Ilya Sutskever “Exploiting Similarities among Languages for Machine Translation” Google Inc., 2013

3. Cho, Merriënboer et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation” Cornell University Library, 2014

4. Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio “Neural Machine Translation by jointly learning to align and translate.”, 2016

5. Cho, Merriënboer et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation” Cornell University Library, 2014

6. Manning, Luong et al. “Effective Approaches to Attention-based Neural Machine Translation” Cornell University Library, 2015

7. Sutskever, Vinyals et al. “Sequence to Sequence Learning with Neural Networks” Cornell University Library, 2015

8. Son, Allauzen et al. “Continuous Space Translation Models with Neural Networks”, Cornell University Library, 2012

9. Papineni, Ward, Zhu “BLEU: a Method for Automatic Evaluation of Machine Translation”, Cornell University Library, 2002

10. Ashish Vaswani, Noam Shazeer et al “Attention Is All You Need”, Cornell University Library, 2017

11. Myle Ott, Sergey Edunov et al “Scaling Neural Machine Translation”, Cornell University Library, 2018