First Phase Report: Semantic Machine Translation (Group 16)

Aditya Prasad Mishra
ASU ID -1211165121
amishr28@asu.edu

Ankit Anand
ASU ID - 1213250712
aanand29@asu.edu

Ashok Prakash
ASU ID - 1214261073
ashokprakash@asu.edu

Laveena Bachani
ASU ID - 1213246721
lbachani@asu.edu

Rohit Bhanushali
ASU ID - 1213179667
rbhanush@asu.edu

1 Motivation

Machine Translation has huge applications in the commercial, political and military domain. The ability to find information regarding any topic should not be affected because of difference in language speaking capabilities [9]. Furthermore, we may not have sufficient data to map one language to another language that we would like to translate [9]. Our system would employ Semantic Machine Translation. It enables us to maintain the grammatical structure of the languages without disturbing literal meaning of sentences [1], unlike Syntax Machine Translation, which destroys the meaning of the sentence. In this project, we will translate English into Hindi and vice versa [3]. To achieve the stated objectives, we will represent the semantics of a language using a Neural Machine Translation Model. Currently, the available NMT models are implemented as Statistical Models. For Phase One implementation, We studied an existing Semantic Model implementation as described in Section 4 below and created our seq2seq model in Section 5 using a Neural Network Model.

2 Problem Formulation

To develop an end-to-end Machine Translation System. We will give input in one language and get the translated output in another.

3 Example

Let us consider a sentence Anna misses her cat; this can be represented in syntax graph, with parts of speech tagging. Anna(NNP) -> misses(VB) -> her (PRP) -> cat (NN). Then, we can convert the resultant graph into the string using the target language graph and dictionary, which will result in a string " एना अपनी बिल्ली को याद करती हे ."

4 Semantic Machine Translation Using Knowledge Graph Model

In this project, we have worked on the understanding of semantic machine translation through a Knowledge Graph model. The basic underlying concept of the given project is HyperEdge replacement grammars. It converts the English language to knowledge graph using K-parser. Then feed the knowledge graph to target language NLgenerator.

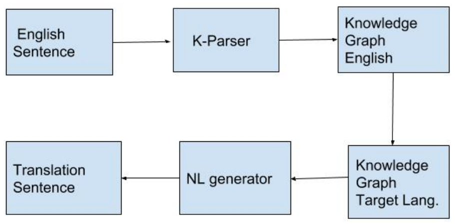

Figure 1: Semantic Translation Flowchart

Algorithm:

4.1 Preprocessing:

Before generating graph does the preprocessing such as:

- **Name Entity Recognition:** Recognizes the entities in the graph.For example when I gave input “Jonh works for IBM at Arizona”. It returns [IBM = ORGANIZATION, Arizona = LOCATION, Jonh = PERSON].
- **Multiword Expression Finding:** It finds the multiple words in the input and joins them with an underscore. For example NEW YORK: NEW_YORK and Ice cream: Ice_cream.

- **Question-Mark Recognition:** If there exists any question mark. Ir removes that from the sentence before parsing and prints later which output. It also then categorize the sentence into interrogative and searches for related question verbs.

4.2 Sentence Graph Generation:

POS Tagging:

The extract method also generates POS tags. POS tags are annotating each word in the sentence with the part of speech tagging, which helps in the lowest level of syntactic analysis and subsequent syntactic parsing and word sense disambiguation.
For our sentence "The Ice cream was eaten by John", generated POS tags are work = VBZ, John = NNP, works - 2 = VBZ, ARIZONA - 6 = NNP, ARIZONA = NNP, act = VBZ, IBM = NNP, person = NNP, etc.

ASP Generation:

After preprocessing it generates the graph for ASP(Answer set representation) to graph parsing node in other language. ASP as Vladimir(2008) states in his paper, "Answer set programming (ASP) is a form of declarative programming oriented towards difficult, primarily NP-hard, search problems. In ASP, search problems are reduced to computing stable models, and answer set solvers(programs for generating stable models) are used to perform the search." [13]
Example: John works for IBM in ARIZONA.

Rules like is_participant_in, is_subclass_of, with_respect_to, in_conjunction_with and prototype_of are figured out in ASP and associated with the words which are annotated from the example text.
The format which generated in the system is given below:

has(IBM-4, is_participant_in, ARIZONA-6).
has(works-2, with_respect_to, IBM-4).
has(works-2, agent, John-1).
has(ARIZONA-6, instance_of, ARIZONA).
has(ARIZONA, is_subclass_of, location).
has(IBM-4, instance_of, IBM).
has(IBM, is_subclass_of, organization).
has(John-1, instance_of, John).
has(John, is_subclass_of, person).
has(works-2, instance_of, work).
has(work, is_subclass_of, act).

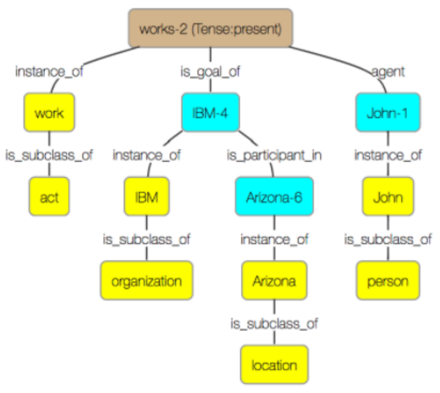

Figure 2: Representation of the knowledge graph for the given example by K-parser

4.3 Translation:

After generating English knowledge graph it is converting to target knowledge graph. For that, it is finding the rule between the left and the right annotated words and translating using the translator.
For example, for our given example it generates following ASP. It is using Glosbe API for translation.
has(इब्म् -4, is_participant_in, अरिज़ोन-6).
has(काम करना-2, VBZ, जोह्-1).
has(काम करना-2, with_respect_to, इब्म्-4).
has(काम करना-2, agent, जोह्-1).

4.4 NLGeneration:

In NL generation it converts the Target Language to the correct semantic representation of the target text.

- **Semantic Corrections:** The verbs are converted to Target Language into its correct form by appending "hai" and "hain" and "kaam " is converted to "kaam karta hai".

- **Verb Disambiguation:** It changes the verb of the source language to the target language. For example: went in the English language with convert to "gaya tha" in Hindi.

- **Helping verb Conversion:** It checks the English helping verbs and converts them to Hindi. For example, if English word contains ”can”. In Hindi, it would be converted to “sakta hai”.

- **Target Language Generation:** In the end it will traverse the Target Knowledge graph in depth-first order and print it.

5 Machine Translation Using seq2seq Model

5.1 Corpus Extraction

Our corpus Extraction program extracts the data from the source and generates the separate .csv files for Hindi and English dataset. After preprocessing, at the end of each sentence/data of datasets, ’EOF’ as a text is inserted to specify that the sentence/instance ends here.

5.2 Model

Attention Mechanism:

Our project employs Attention Mechanism. This mechanism allows the decoder to attend to different parts of the source sentence at each step of the output generation. This can be achieved by automatically generating a context vector for each output time step and not a single fixed content vector. Decoder easily generates the output for i’th timestamp by looking into the i’th context vector and the previously hidden outputs. [10]

Embedding:

We are performing embedding on inputs to map the words to a vector of real numbers. We split the text into words and assign each word in the vocabulary an integer value. Words which are not present get converted to ’unknown’ and are assigned the same embedding. Learning the embeddings from scratch is useful given the size of training data present. [11]

Encoder:

Encoder contains a single Basic LSTM cell.
It has Dropout with the probability of 0.5 (50%).
Multi RNN cells are present for encoding the input depending upon the number of layers present.

Decoder:
Multi RNN cells are there for decoding the input. Dropout with a probability of 0.5 (50%) is present. It contains one Basic LSTM cell. Decoder returns the training predictions and test predictions after processing data and decoding training and test set.

5.3 Training

For Training purpose, we have set the value of epoch to 100, batch size to 64 and learning rate is set to 0.0001. We train our model by passing these parameters. For now, we have six layers(3 encoder layers, 3 decoder layers) in our model.
We have added dropout equal to 0.5 (50%) in our model to prevent overfitting of the data. There is a possibility that we will get the highest variance when dropping a neuron with 0.5 probability.
Initially, Input sentence converts to word2vec. For better training and test accuracy, we are also performing cross-validation on the training dataset.

6 Future Work

- We are currently evaluating our model, and it will be ready for Phase 2.
- Our Model is currently using Statistical parameters; We will represent Semantics in Phase 2.
- We are also trying to integrate Voice module into our project, which in turn will enable us to get input via voice/speech.

7 Related Work

- We have refered to [14] for our intial model.
- We have taken some help from TensorFlow NMT (seq2seq) tutorial. [11]

References

[1] Jones, Bevan, et al. ”Semantics-Based Machine Translation with Hyperedge Replacement Grammars.” COLING. 2012.

[2] Chen, Yun, Yang Liu, Yong Cheng, and Victor OK Li. ”A Teacher-Student Framework for Zero-Resource Neural Machine

Translation.” arXiv preprint arXiv:1705.00753 (2017).

[3] https://www.cse.iitk.ac.in/users/cs671/2015/ _submissions/egupta/project/report.pdf.

[4] papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[5] https://arxiv.org/pdf/1409.1259.pdf

[6] https://arxiv.org/pdf/1406.1078.pdf

[7] https://nlp.stanford.edu/pubs/luong2016iclr _multi.pdf

[8] https://cs224d.stanford.edu/papers/nmt.pdf

[9] https://pdfs.semanticscholar.org/b3e8/ 9f05876d47b9bd6ece225aaeee457a6824e8.pdf

[10] https://hackernoon.com/attention -mechanism-in-neural-network-30aaf5e39512

[11] https://www.tensorflow.org/tutorials/seq2seq

[12] Piotr Chabierski, et al. ”Logic-based Approach to Machine Comprehension of Text.”Student-projects-ug-2016-17. 2017

[13] https://www.cs.utexas.edu/users/vl/wiasp.pdf

[14] https://www.superdatascience.com/courses/deep-learning-and-nlp-a-z