REPORT

EDS-6397: Natural Language Processing

Bilingual Spanish-English and English-Spanish translator using BiLSTM with Attention.

Submitted by:
Varun Vaddi (2347481)

Vamsi Krishna Yerubandi (2351142)

Contents

1. Introduction

Problem Statement:

The goal of this project is to develop a bilingual Spanish-to-English and English-to-Spanish machine translation system using Long-Short Term Memory (LSTM) Network. This system will automatically translate between Spanish and English. Machine translation has become an essential tool in breaking down language barriers, and the main aim of this project is to create an efficient and accurate translation model capable of handling both languages (Spanish and English) in both directions. The main challenge lies in capturing grammar, subtle distinction and context of both languages, while ensuring the translations are fluent and contextually accurate. English and Spanish differ in many linguistic aspects, such as word order (e.g., Subject-Verb-Object in English vs. Subject-Object-Verb in Spanish), gendered nouns, verb conjugations, and the use of articles.

Literature Review:

In recent years, significant progress has been made in the field of neural machine translation (NMT), particularly using recurrent neural networks (RNNs) [6], including Long Short-Term Memory (LSTM) [7] networks, which are designed to handle sequential data and mitigate the vanishing gradient problem associated with traditional RNNs.

Key Approaches in Machine Translation:

1. **Sequence-to-Sequence Models (Seq2Seq)** [2][10]**:**
   Early methods used **encoder-decoder** models, where the encoder turns input sequences (like Spanish sentences) into a fixed-length vector, and the decoder generates the output (like English translations). While these models were effective, they struggled with longer sentences because the fixed-length vector often lost important details.

2. **Bidirectional LSTM (BiLSTM)** [3]**:**
   BiLSTM networks improve on regular LSTMs by processing input sequences in both directions (forward and backward). This helps capture more context, which is especially useful in languages like Spanish and English, where word order can be flexible and meaning depends on context from both ends of the sentence. This makes BiLSTM better at handling ambiguity and long-range dependencies.

3. **Pretrained Models and Transfer Learning:**
   Recently, large pretrained models like **GPT-3** [4] and **BERT** [5] have made a big impact on machine translation. These models are trained on massive amounts of data and can be fine-tuned for specific tasks, making them good at translation. However, training them requires a lot of computational power and data.

2. Dataset Used

For this project on Bilingual machine translation (Spanish-to-English and English-to-Spanish), a parallel corpus of Spanish and English pairs from Tatoeba Dataset is used. Tatoeba dataset is widely recognized open-source collection pairs in various languages.

Description of the Dataset:

- **Source**: The data is sourced from the Tatoeba project [1] which is a collaborative effort to create a collection of sentences and their translations. The specific subset of this collection used for this project includes translations between Spanish and English. The dataset is available for free and updated regularly.
- **Size**: The Tatoeba English-Spanish subset used in the project contains 141,543 sentence pairs. The sentences are typically short and contain a variety of structures, making it suitable for machine translation models.
- **Type**: The data is textual, with each sentence being a sequence of words in either language. Each sentence pairs consists of an English sentence and its corresponding Spanish translation
- **Preprocessing**: Several preprocessing steps are applied on Tatoeba dataset to prepare it for training our Machine translation model.
    - **Shuffling:** Used Random shuffle function from NumPy [13] library to rearrange the order of sentence pairs in the dataset. Randomizing the order helps the model learn to generalize better, and not just memorize the order of the data and prevents Overfitting.
    - **Stripping Whitespaces:** Used strip function to remove both leading and trailing spaces from Target sentence.
    - **Addition of Special tokens:** This is a common preprocessing step in Sequence-to-Sequence (Seq2Seq) models. Added start token (\t) and end token (\n) to mark the beginning and ending of a sentence. These tokens help the model understand when to start generating the translation and when to stop.
    - **Tokenization**: Tokenization is the process of converting text data into numerical data which can be processed by the model. TensorFlow's keras.preprocessing.text module to tokenize the text at Word level. Used **<unk>** as OOV token, which will be used to replace any word that was not seen by the model during training. Tokenization is applied on both input and target sentences separately. Vocabularies for both languages are created.
        - **Max Spanish Sequence Length** is 68.
        - **Max English Sequence Length** is 72.
        - **Spanish Vocabulary Size** is 40,706.

        - **English Vocabulary Size** is 23,225.
    - **Fitting on texts**: This function fits the tokenizer on the English and Spanish sentences, thereby mapping each word of the dataset to a unique integer.
    - **Converting Texts to Sequences: This** function converts each word in the input and target sentences into its corresponding integer from Tokenizers Vocabulary, Thus the sentences are transformed into sequence of numbers.
    - **Padding:** Since the sentences vary in length, they need to be padded to the same length to ensure the model receives consistent input. So, we apply padding to both input and target sequences. This padding is added at the end of sentences.
    - **Embeddings:** For this project, we used Google News Word2Vec [8] model. This model is trained on a large corpus of news articles, and it contains word vectors for 3 million words and phrases. Each word is represented by a 300-dimensional vector.
    - **Creation of Embedding matrix:** An Embedding matrix [14] is created, which is a 2D array where rows represent words and columns represent dimension in vector space. This matrix helps the model during training to look up the vector for each word.
        - The embedding matrix is initialized with random values between -0.1 and 0.1.
        - Added on extra row with index 0 for Padding token.
        - Then, if the word is found in Word2Vec [8] model, then its corresponding word vector is copied into the appropriate row in the embedding matrix.This Embedding matrix is then fed into encoder and decoder of Machine translation model.
        - Embedding matrix size for Spanish is (40706, 300) & for English is (23225, 300)

3. Methodology

We used a BiLSTM-based Sequence-to-Sequence model [9] with Attention is used for Bilingual Translation between Spanish and English. This Model contains two main components Encoder and Decoder. The Encoder processes the input sequence, while the Decoder generates the output sequence. We also incorporated Attention mechanism, as it improves translation quality by allowing the decoder to focus on specific parts of the input sequence at each step.

Encoder Architecture:

- The Encoder processes the input sequence using a BiLSTM layer, thereby creating a context vector and hidden states that represent the sentence which will guide the translation in the decoder.

- **Input Layer:** This layer accepts sequences of different lengths as input, so this shape is not fixed.
- **Embedding Layer:** This layer converts word indices into dense word vectors using Word2Vec. It uses the encoder_embedding_matrix as weights, the trainable is set to True, which allows for fine-tuning of embeddings during training.
- **Bi-directional LSTM Layer:** This layer processes the embedded input in both forward and backward directions.
    - This layer has 256 neurons. Used L2 regularization [11][12] to keep the weights small and prevent overfitting, we added an L2 regularization of strength 0.01.
    - The return_sequences is True, ensures that the LSTM returns full sequence of outputs at each step. The return_state is True, ensures that the final hidden(h) and cell states(c) are returned.
- **Concatenation**: This layer is used to combine the forward and backward hidden states(state_h) and cell states(state_c) are concatenated to form the final context vector.
- **Encoder model:** This layer takes encoder_inputs and returns encoder_outputs along with final hidden(state_h) and cell states(state_c).

Decoder Architecture:

- The Decoder generates the output translations word by word using an LSTM, with the help of encoded context provided by the encoder. We also use the Attention mechanism to focus on relevant parts of the input sequence during the translation process.
- **Input Layer:** This layer accepts the previously generated tokens from Decoder, starting from the special token.
- **Embedding Layer:** This layer is similar to the Embedding layer in Encoder, it transforms the word indices into dense word vectors. This layer uses decoder_embedding_matrix that was created earlier as weights.
- **Decoder state inputs: It** is the list of the two states hidden state and cell state allow the decoder to retain the memory to generate the translated output.
- **Decoder LSTM:** The LSTM in decoder processes the embedded target sequences and uses the context vector along with the hidden states from the encoder in generating the output.
    - This layer has 512 neurons. Used L2 regularization [11][12] to keep the weights small and prevent overfitting, we added an L2 regularization of strength 0.01.
    - The return_sequences is True, ensures that the LSTM returns full sequence of outputs at each step. The return_state is True, ensures that the final hidden(h) and cell states(c) are returned.

Attention Mechanism:

- Attention mechanism allows the Decoder to focus on the most important words from the input sentence at each step. This layer computes the context vector by comparing the decoders output sequence with the encoder output sequence. Then we concatenate the decoder output with the context vector to provide enriched context

Output Dense Layer:

- This layer is a Dense layer with a SoftMax activation function that produces probability distribution of the target translated sequence. We have added an L2 regularization here. The word with highest probability is selected as the next word in translated sentence.

Decoder model for Inference:

- The model accepts the input word, context vector, hidden and cell states, thus producing the predicted word and updates the hidden and cell state values that are used in the next step.

Encoder and Decoder Validation:

- The encoder and decoder models are validated using test inputs to ensure they function correctly using a single test sample.
- We use a dummy target variable, which is initialized with zeros, which represents a start token.

Compute the attention function:

- This function computes the attention scores to generate the context vector. The context vector is a weighted sum of encoders output, where the weights are determined by the context vector.
- The first step is to compute the attention scores, this is computed for each pair of encoder output and decoder output using dot product. This measures the similarity between the current decoder's state and each of the encoder's state.
- Then, the attention scores are normalized using SoftMax activation function, which converts the scores into probability distribution. The attention weights for each decoder step will sum up to 1

4. Results

4.1 Spanish-to-English translator

In this section, we present the performance of Spanish-to-English Translation model and evaluate its output using the below metrics.

Model Output:

- **Training/ Validation Accuracy and Loss:**

```
Epoch 26/50
1375/1375 ━━━━━━━━━━━━━━━━━━━━ 113s 82ms/step - accuracy: 0.9655 - loss: 0.3320 - val_accuracy: 0.9570 - val_loss: 0.3906
```

- **Test Translations:**

```
Input Sentence (Spanish): Eso no puede ser correcto.
Predicted Translation (English): that can't not right.
Actual Translation (English):    That can't be right.
```

```
Input Sentence (Spanish): Necesito saber algo.
Predicted Translation (English): i need to to to
Actual Translation (English):    I need to know something.
```

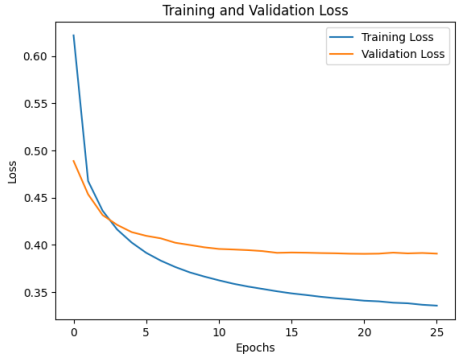
```
Input Sentence (Spanish): ¿En qué tren vino usted?
Predicted Translation (English): what what did you you you you?
Actual Translation (English):    What train did you come on?
```

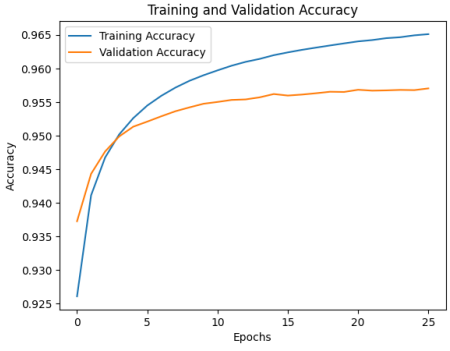
Evaluation Metrics:

- **BLEU (Bilingual Evaluation Understudy) Score** [15]:
  - Avg BLEU Score on Test data: 0.0485
- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score** [16]**:**
  - ROUGE-1 (Unigram Recall): 0.4107.
  - ROUGE-L (Longest Common Subsequence): 0.4040.
- **Test Accuracy and Loss:**
  - Test Loss: 0.345.
  - Test Accuracy: 96.35%.

```
Test Loss (Spanish to English): 0.34507232904434204
Test Accuracy (Spanish to English): 0.9635195136070251
```

- **Visualization of Model Performance:**





4.2 English-to-Spanish translator

In this section, we present the performance of English-to-Spanish Translation model and evaluate its output using the below metrics.

Model Output:

- **Training/ Validation Accuracy and Loss:**

```
Epoch 10/50
1000/1000 ━━━━━━━━━━━━━━━━━━━━ 89s 89ms/step - accuracy: 0.9644 - loss: 0.1339 - val_accuracy: 0.8036 - val_loss: 1.4590
```

- **Test Translations:**

```
Input Sentence: I need to finish my homework.
Predicted Translation: necesito estirar mi tarea.
Actual Translation:     Necesito terminar mi tarea.
```

```
Input Sentence: I need to hire an accountant.
Predicted Translation: necesito contratar un esquíes
Actual Translation:     Necesito contratar a un contador.
```

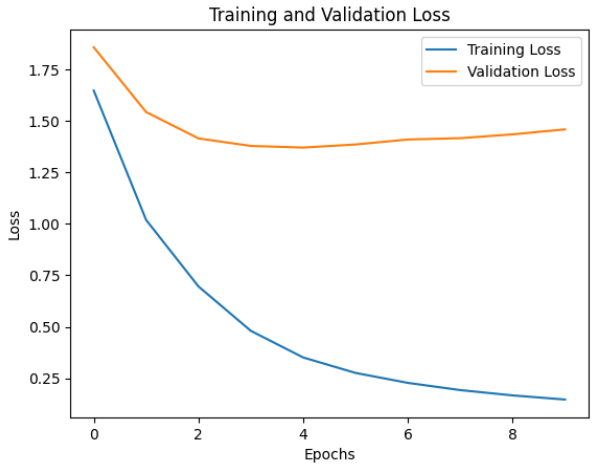
```
Input Sentence: I need to go to the bathroom.
Predicted Translation: necesito ir al baño.
Actual Translation:     Necesito ir al baño.
```

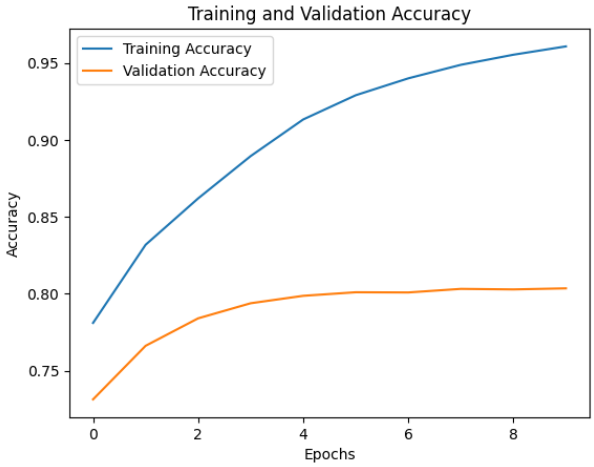
Evaluation Metrics:

- **Test Accuracy and Loss:**

```
Test Loss: 1.420251488685608
Test Accuracy: 0.7920187711715698
```

Visualization of Model Performance:





5. Discussion

5.1 Spanish-to-English translator

Evaluation Metrics:

- **BLEU score** [15] evaluates the overlap between n-grams in the predicted translations and predicted translations.
    - The Avg BLEU score is relatively low (**0.0485**). This score suggests that although the model produces translations that are accurate, it often fails to match every word to word in the same order as reference translations. This could be due to the differences in Spanish and English languages.
- **ROUGE score** [16] focuses primarily on Recall, this score indicates how well the models predict output captures the relevant context.
    - ROUGE-1 (Unigram Recall): **0.4107**, indicates a good recall of individual words in translations, but there is still some room for improvement.
    - ROUGE-L (Longest Common Subsequence): 0.4040, indicates a reasonable sentence structure retention, but there is still some room for improvement.
- **Test Loss: 0.345**, indicates the loss is relatively low suggesting that the model is performing well at minimizing the error in test dataset, it means that the predictions are quite close to actual predictions.
- **Test Accuracy: 96.35%** is quite high, which indicates that over 96% of the time, the predicted translations are the same as reference translations.

Challenges and Limitations:

- **Overfitting:** one challenge we faced is overfitting during training, where models performed well on training data but struggled during validation. On close inspection, we found that this is due to the data being sorted in such a way that similar contextual sentences are placed together in order. We overcame this overfitting issue by randomly shuffling all the pairs during preprocessing.
- **Sentence Complexity:** The model may struggle with complex sentences or long-range dependencies in the input text.
- **Vocabulary Mismatch:** The model may not perform as well on sentences containing rare or out-of-vocabulary words, especially if the model hasn't seen these words seen during training. This can lead to the generation of incorrect translations.One method to get around this is to increase the training size so that it covers a lot of vocabulary with rich contextual information.

Suggestions for Future Work:

- **Data Augmentation:** We can perform data augmentation to expand the training dataset to improve models' generalization capabilities.
- **Synonym replacement:** In this process, we substitute some words with their contextually similar synonyms, this helps in reduction of Out-of-Vocabulary words.
- **Stacking LSTM layers:** we can also stack LSTM layers in encoder and decoder parts of the model might improve the model's ability to capture complex patterns in input sentences.

- **Multi-head Attention** [17]**:** Incorporating multi-head Attention could enhance the ability of the model to focus on multiple parts of the input sentence at the same time, rather than focusing on single part.

5.2 English-to-Spanish translator

Interpretation of Results:

- **Strengths**:
  - The model correctly captures the general intent in some cases (e.g., “necesito” for “I need”).
  - Basic sentence structures are understood, and translation starts with the correct word.

- **Weakness:**
  - **Word-Level Errors**: The model struggles with specific word translations, such as “child” being incorrectly translated to “momento” instead of “niño.”
  - **Context Misinterpretation**: Sentences with specific contexts (e.g., “homework,” “accountant”) result in unrelated words (e.g., “lapicera,” “esquíes”).

Evaluation Metrics:

- **Test Loss: 1.42** demonstrates that the model has achieved a moderate level of error in its predictions.
- **Test accuracy: 79.2%** indicates that the model correctly predicted nearly 4 out of 5 sequences, and the translations are also on point.

Challenges and Limitations:

- Overfitting: Initially, before shuffling the data when I trained the model is highly overfitted as the data is in a serially increasing fashion and thus later, we shuffled the data first and then trained to prevent overfitting and added a dropout of 0.5 as well.

Suggestions for Future Work:

- **Data Augmentation:** Use paraphrasing techniques to create variations of English sentences, enriching the dataset with diverse sentence structures.
- **Transformer Architecture**: Consider transitioning to Transformer-based models like BERT or GPT, which handle long-term dependencies and parallel computations more efficiently than BiLSTMs.
- **Layer Normalization**: Normalize activations within layers to stabilize training.

6. Conclusion

The BiLSTM-based Sequence-to-Sequence model with attention mechanism performs has demonstrated varying level of performance across Spanish-to-English and English-to-Spanish translation tasks. The

Spanish-to-English model achieved a relatively low BLEU score (0.0485), indicating room for improvement in translating word order and sentence structure. However, it performed well in terms of ROUGE-1 (0.4107) and ROUGE-L (0.4040), suggesting good word recall and sentence structure retention. The test accuracy was high at 96.35%, showing that the model accurately predicts most of the translations.

On the other hand, the English-to-Spanish model exhibited moderate test accuracy (79.2%) with some word-level errors and context misinterpretation, highlighting challenges in specific vocabulary and context handling. Main challenges faced included overfitting and handling complex sentences, especially when sentences contained out-of-vocabulary words. These limitations were addressed with techniques like data shuffling and dropout during training. For future work, we suggest using data augmentation, synonym replacement, and exploring advanced techniques like stacking LSTM layers and multi-head attention to improve model performance further.

7. References

- [1] Tatoeba Project Dataset https://tatoeba.org/en
- [2] Luong, M. T., Pham, H., & Manning, C. D. (2015) "Effective approaches to attention-based neural machine translation." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.
- [3] Cho, K., Merrienboer, B. van, Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014) "On the properties of neural machine translation: Encoder-decoder approaches." *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.
- [4] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shinn, A., Bhor, R., & Wu, J. (2020). Language models are few-shot learners. *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*.
- [5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pretraining of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*
- [6] Bengio, Y., Simard, P. A., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*.
- [7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation.*
- *[8] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.*
- *[9] Cho, K., Merrienboer, B. van, Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches.*

Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation.

- *[10] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Proceedings of NeurIPS 2014.*
- *[11] Ridge, A. E. (1970). The Use of Regularization in Multivariate Data Analysis. Springer-Verlag.*
- *[12] Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. Proceedings of the 21st International Conference on Machine Learning (ICML 2004).*
- *[13] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., & others (2020). Array programming with NumPy.*
- *[14] Goldberg, Y., & Hirst, G. (2014). Exploring the nature of word embeddings. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*
- *[15] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002).*
- *[16] Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. Proceedings of the Workshop on Text Summarization Branches Out.*
- *[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. A., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Proceedings of NeurIPS 2017.*
- *[18] Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.*