**Interim Report: NLP Capstone Project – Machine Translation**

# 1. Problem Statement

The task is to design a Machine Translation model that can be used to translate sentences from one language to another. The database to be used comes from ACL2014 Ninth Workshop on Statistical Machine Translation. This workshop mainly focusses on language translation between European language pairs. The idea behind the workshop is to provide the ability for two parties to communicate and exchange the ideas from different countries.

# 2. Data Description

The database we will use has sentences in German/English of various events. Three such datasets have been obtained from the Statistical Machine Translation workshop. The datasets are available at [this link](https://statmt.org/wmt14/translation-task.html) and comprises [Europarl v7](https://statmt.org/wmt13/training-parallel-europarl-v7.tgz), [Common Crawl Corpus](https://statmt.org/wmt13/training-parallel-commoncrawl.tgz), and [News Commentary](https://statmt.org/wmt14/training-parallel-nc-v9.tgz).

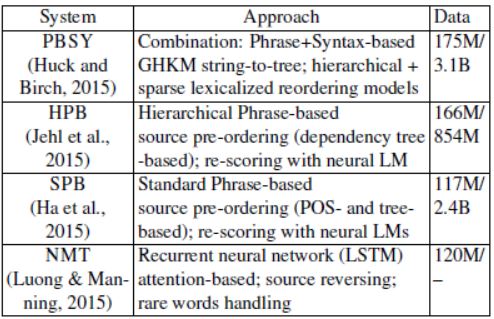
# 3. Project Objective

In this project, we will focus only on translating English sentences to German.

# 3. Research

Before embarking on implementing a solution to our problem, we looked at a few research papers and available literature/methodologies on machine translation between different languages. In this context, we looked in detail at [Neural versus Phrase-Based Machine Translation Quality: A Case Study](https://arxiv.org/pdf/1608.04631.pdf) and [Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation](https://arxiv.org/pdf/1609.08144.pdf). We learned the following points from our study.

1. English-German language translation is known to be particularly hard because of morphology and syntactic differences.
2. There are various translation systems that exist today (both Phrase-Based and Neural Translation) some of which have been highlighted in the table below.



1. Existing tools for translation error detection are either based on Word Error Rate (WER) or Position-independent word Error Rate (PER). TER (Translation Edit/Error Rate) is a method used by machine translation specialists to determine the amount of post editing required for machine translation jobs.
2. Regarding error classification, [Hjerson](https://ufal.mff.cuni.cz/pbml/96/art-popovic.pdf), an open-source tool for automatic error classification of machine translation output, detects five main types of word level errors.
   1. Morphological
   2. Re-ordering
   3. Missing words
   4. Extra words
   5. Lexical choice errors
3. Neural Machine Translation (NMT) is an ensemble of 8 long short-term memory (LSTM) networks of 4 layers featuring 1,000-dimension word embeddings, attention mechanism, source reversing, 50K source and target vocabularies, and out of vocabulary word handling.
4. NMT systems lack robustness when input sentences contain rare words.

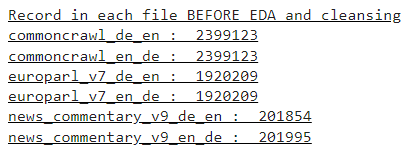
**Although the NMT System is computationally costly and resource demanding, it outperformed Phrase-Based Machine Translation (PBMT) systems on English-German translation according to the International Workshop on Spoken Language Translation (IWSLT) 2015 evaluation campaign. Therefore, we selected NMT for our machine translation approach.**

We also looked at a few papers that highlighted pre-processing techniques and methodologies. One notable among them is [Pre-Processing of English-Hindi Corpus for Statistical Machine Translation](https://www.scielo.org.mx/pdf/cys/v21n4/1405-5546-cys-21-04-00725.pdf). We made some choices for pre-processing based on our findings.

1. ***Casing*:** In English, capitalization is used in the beginning of sentences, to indicate a named entity or a proper noun. This, in turn, may help to facilitate part-of speech (POS) tagging and Named Entity Recognition (NER). However, capitalization may degrade performance of statistical machine translation, as the occurrence of a word with and without capitalization would be treated as two different words. The extent of performance degradation would mostly depend on how many such words are found in the corpus that have both lower-case and upper-case variants. Another aspect to note is that the capitalized words in proper nouns such as “New” in “New Delhi” need to be treated differently than its lowercase counterpart such as “new”. Since we decided to keep such a distinction and the fact that our vocabulary selection would only be a subset of the entire dataset for modeling and translation, we kept the original casing for both English and German languages.
2. ***Punctuations*:** Punctuation is a mechanism for putting emphasis and for clarity of expression. It helps the reader in terms of readability of an expression. Additionally, punctuation becomes very important for conveying the intended meaning of an expression, as the placement of punctuation marks also helps in disambiguation of an expression. Therefore, for machine translation activity, punctuations have been retained so as to not dilute the context or the meaning on which a sentence is based.
3. ***Numbers*:** Numbers appear quite frequently in corpus and generally do not contribute towards translation and make the phrase table noisier. However, these need to be passed to the translations for fluency and transfer of information. Therefore, we have retained numbers as well.
4. ***Named Entities*:** Named Entities are expressions that appear quite frequently in text and are much more variable in nature than content words. In the translation process, these generally should not get translated and appear in transliterated forms which are commonly phonetic representations in the target language. The appearance of different Named Entities in corpus presents difficulty in learning their translation, as these may be unknown to the training corpus (being an Out of Vocabulary i.e. OOV term) or not having sufficient appearances to learn their translation reliably. A [Stanford study](https://nlp.stanford.edu/courses/cs224n/2010/reports/singla-nirajuec.pdf) shows that treating Named Entities differently does help in providing better Machine Translations and improves human readability overall. Given the reduced scope of our work for academic purpose and the fact that the BLEU score, which we use here for translation efficiency, did not vary much with special treatment of the Named Entities as per the study, we decided not to do any additional processing of the Named Entities.
5. ***Normalization*:** Spell normalization is a process by which text is transformed in some way to make it consistent in terms of usage of spellings for its words throughout the text. The intention behind this activity is to reduce the lexical redundancy. Since lemmatization is not meaningful for machine translation purposes, we have not normalized the dataset.

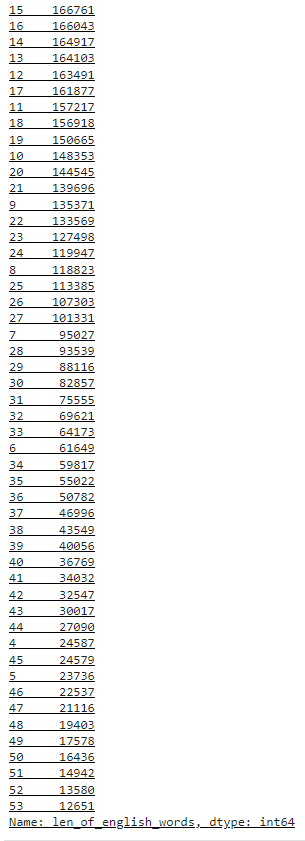
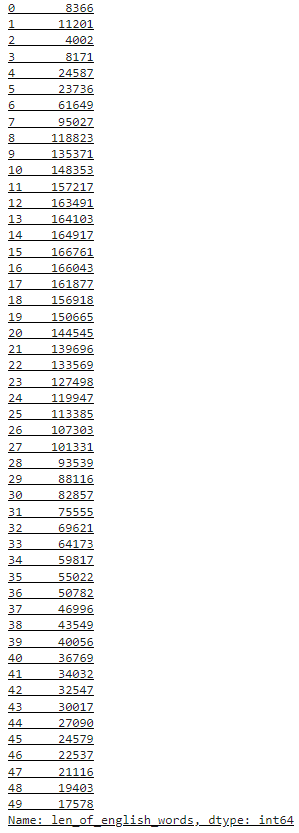
# Exploratory Data Analysis (EDA) and Pre-Processing

We used Google Colab for all our runs and analysis. We read in the 3 datasets – Europarl v7, Common Crawl Corpus, and News Commentary. We looked at the number of records in each of those datasets.



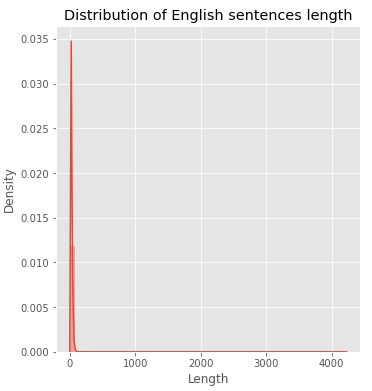
The Common Crawl and Europarl v7 have matching numbers of English and German entries. However, the News Commentary dataset has an unequal number of English and German entries. We analyzed the News Commentary dataset. There are 141 more entries of English than that of German. We printed some random entries and observed that some of the translations, although present, are not in their correct positions, i.e. there is no one-to-one mapping between the English entry and its corresponding German entry or vice versa. We also noticed that some of the translations are incorrect in that they refer to sentences with completely different meanings (as per Google translate). Given these challenges, we decided to drop the News Commentary dataset and merge only the other two datasets.

Further processing was done on the merged dataset. We looked at the length of words in each sentence and sorted the distribution based on word count. Then, we looked at the frequency of word counts across sentences in the dataset. The top 50 data points sorted by word count as index is shown in the figure on the left-hand side. We also looked at which word count had the highest frequency across sentences as shown in the figure on the right-hand side.

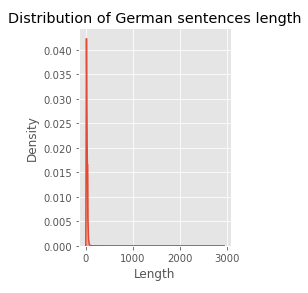


So, there are 8366 empty sentences, 1 word in 11201 sentences, 2 words in 4002 sentences, and so on. Also, we see the maximum number of sentences, i.e. 166761, have a word count of 15, followed by a word count of 16 in 166043 sentences, and so on. The greatest number of sentences are concentrated in the word count range of 8 to 27.

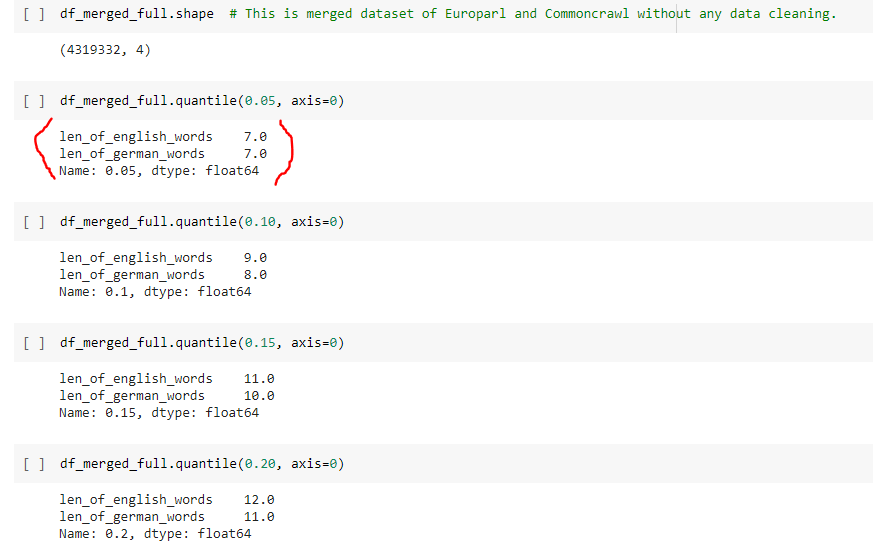
The plot of the word counts for English sentences reveals that the majority of the English sentences have 0 to 100 words.



Similarly, the plot of the word counts for German sentences reveals that the majority of the German sentences have 0 to 100 words.



To make it computationally efficient, we decided to take only a subset of the dataset (i.e. 5%) for modeling purpose. We looked at the quantile information to determine what might be a good subset of the data to use.

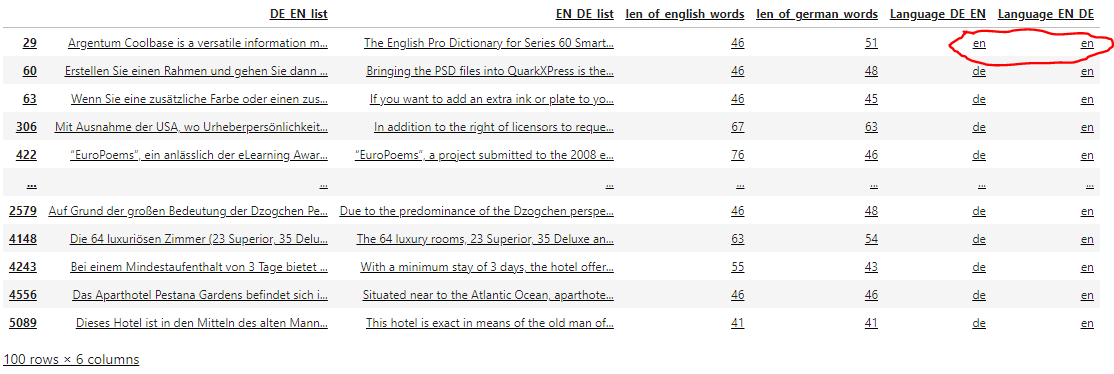


From the quantile information we can see that roughly 5% of the data lies with length of words <= 7 in both languages. We decided to take only the English sentences with 1 to 7 words in both English and German languages. This allowed us to scale down the dataset from 4.32 million entries to 161K entries. This assumption is made because longer sentences tend to express concepts, ideas and opinions causing translations to vary based on the style and understanding of the translator. Given such circumstances, not only the translation task becomes significantly complex, but also there is a good chance the training dataset may not be of the necessary quality. Since this is an academic project with a strict timeline and limited computational power, smaller sentences are intentionally chosen to help in learning the basic nuances of translation.

In this context it is worthy of mentioning that, since we have used a small sample data with sentences equal to or less than seven words, stop words which occur in all sentences usually, have seemed to overwhelm the model learning process. We have observed that the predicted or translated sentences are having more than usual stop words.

Hence, after consulting our mentor we have decided to continue the model training process using an extract from the original dataset that represents the population from the length of sentences perspective. The team intends to gain some new insights related to the challenges in translating longer sentences from an academic point of view.

Next, we applied the language detection utility (langdetect) to check if the sentences in English were really being categorized as English and if those in German were really being categorized as German. We found that this was not the case with many entries. There were cases where a sentence in the German dataset was being categorized as an English one as highlighted in the first row (index 29) in the figure below.



We decided to do away with all those entries which did not have English to German translations and kept only the ones that had English to German translations as our final dataset for feeding as input to the Modeling step. The final dataset, therefore, was left with 129K entries.

# Model Readiness: Summary of Assumptions Based on EDA and Pre-Processing

1. There are 2,399,123 rows found in the Common Crawl dataset, one for English and one for German language. Similarly, there are 1,920,209 rows found in the Europarl-v7 dataset, one for English and one for German language. It is assumed that the translation for any English line with, say, index number X, will be present at the same index location X in the German file, and vice versa. Although some random tests are done by the project team using google translator, it is important to make this assumption because it is not practical to verify all 4 million rows. In real production scenarios, this assumption could pose significant risk. Only datasets with a high confidence level should be used.
2. In real scenarios, both German language experts and English language experts should verify the quality of translations in the datasets. Since there are no German language speakers in our project team and none of us are linguistic experts, it is assumed that the translation provided in the datasets is of decent quality suitable for translations.
3. Based on the project problem narrative, the project team will focus only on English to German translation. German to English translation is not considered in the scope of this project.
4. There are many instances where the number of lines in a row stretch into more than hundreds of sentences and thousands of words. It is assumed that the corresponding translation also has a similar structure which is learnable by the Machine Translation models.
5. Although the files are distinctly tagged as either English or German, several lines are found in both the files which are neither English not German. To rectify this discrepancy, first, the datasets are horizontally merged to preserve their translation alignment. Subsequently, a language detection function is run separately on each row in each language column to tag it separately with the detected language. There are instances where the rows are empty. Those rows are tagged as ‘Unknown’. To expedite the process only a few first words are used to detect the language of any row.

Finally, after tagging the rows on either side with languages, if any one line – say on the English side – is found to be not English, then that line along with its corresponding German line is deleted. This helps to preserve the one-to-one correspondence in translation between the English and German datasets. This exercise is repeated for German columns as well. The remaining lines, both English and German are considered to be of decent quality without grammatical errors, with similar verbosity, etc.

1. In the News Commentary dataset, while the German dataset consists of 201,854 rows, the English counterpart consists of 201,995 rows. A number of tests have been performed to check if there is any ‘kind’ of alignment between the translations but the project team could not establish such evidence. German lines were translated using Google translator into English to check if there is any simple alignment issue which could be rectified. However, the project team failed to identify any dependable alignment.

Since the number of rows present in the News Commentary dataset is significantly lower compared to that present in the ‘Europarl-v7’ and ‘Common Crawl’ datasets put together, it is assumed that there will not be any significant loss in accuracy if ‘News Commentary’ dataset is not used for training and testing purposes. The project team has performed extensive search on the web as well to see if any documentation is present relating to this issue but failed to find any constructive material. Hence, to be on the safe side, the ‘News Commentary’ dataset has been ignored.

1. Since the project team is using very limited computational resources – even if Google Colab is used – it is taking a lot of time to pre-process the 4 million translations that are available in Europarl-v7 and Common Crawl datasets together.

Given that this project is mainly for academic purposes, only a portion, i.e. 5% corresponding to the lower quantile of word count data in sentences from the merged dataset, will be used for training and testing the models. It is understood that usage of small extracts of data may negatively impact the final performance that can be achieved for a given model.

If model A performs better than model B when using smaller extracts of the datasets, it is assumed, in the context of this project, that the same difference in performance will be observed when all 4 million+ translations are used for training. However, it is quite well understood that this assumption may not hold true as the variety in the smaller set may not be the same compared to that in the full dataset.

1. It is assumed that the Europarl-v7 and Common Crawl datasets together consist of the requisite variety and volume of translations required for training a machine translator satisfactorily.

# Model Selection

The team was assigned the task to build three mandatory models and one optional model on the merged dataset after initial preprocessing. We have developed each of the mandatory models as highlighted under the ‘Model Building’ section.

The mandatory models are:

1. **Model 1:** Design, train and test simple RNN & LSTM model
2. **Model 2:** Design, train and test RNN & LSTM model with embeddings
3. **Model 3:** Design, train and test bidirectional RNN & LSTM model

The optional model is:

1. **Model 4:** Design, train and test Encoder-Decoder RNN & LSTM model

# Model Building

## Model 1: Simple RNN LSTM Model

A simple RNN LSTM model is built without explicit embedding outside the network. The word embedding is done by a layer with dimension equal to the size of vocabulary, very similar to one hot encoding. While one layer encodes the input language, another layer after repeater vector will decode into the target language. The number of LSTMs in each layer is kept same but as a variable. The learning is monitored and measured using RMS optimizer.

## Model 2: LSTM model with Embeddings

An LSTM model is implemented with word embeddings from glove.6B.100d.txt (GloVe). GloVe is an unsupervised learning algorithm to learn vector representation, i.e word embedding for various words. GloVe stands for Global Vectors for Word Representations. In this model, we have used the 100-dimensional GloVe vectors for the machine translation task. All the words which are not in the GloVe dictionary are being assigned to zero vector.

The sequential model starts with an Embedding layer with weights from embedding matrix from GloVe file. Output dimension of this layer is kept as 100 to sync with 100-dimensional GloVe vector. Embedding layer is followed with couple of LSTM layers with dropout of 20% to avoid overfitting. A RepeatVector layer is used in between the LSTM layers to repeat the inputs. Final layer is to convert the output to dense encoding with activation function used as ‘softmax’.

Model 3: Design, train and test bidirectional RNN & LSTM model

The bidirectional model uses an initial embedding layer followed by 2 bidirectional LSTM layers each with 512 neurons. A dropout of 10% is used to reduce overfitting. The final dense layer uses a softmax function. RMSprop is used as the optimizer. The model was run for 100 epochs with early stopping criteria as the ‘validation\_loss’.

Model 4: Design, train and test Encoder-Decoder RNN & LSTM model

In this encoder decoder model, we have attempted to read the data set (5000 lines) considering the training hours and resource limitations (System RAM/GPU). After reading the data the target language is appended with “START” & “END” tags. Since the dataset is already pre-processed, no further pre-processing is done. Then the following is done for source & target language,

1. Vocabulary building
2. Ascertaining the maximum sentence length
3. Preparing the encoder/decoder tokens, encoder/decoder input data, decoder for target data, token index for input & output text

After completion of above steps, we proceeded to build the model with Encoder/decoder for input LSTM layer with an embedding layer and a dense decoder output layer. This model was trained with an Embedding size of 120 and 324 LSTM units were used.

# Model Evaluation

## Model 1: Simple RNN LSTM Model

The model is a very simple one compared to the state of the art in machine translation models. Due to its simplicity, it is not expected to perform highly compared to even the other models developed as part of this model. However, the model has provided valuable insights regarding the challenges of translating one language into another. We got an accuracy of ~63% and BLEU score in the range of 0.11 to 0.32 for 4-grams.

## Model 2: LSTM model with Embeddings

The use of embedding vector from GloVe is expected to give good coverage of English words. The model with 2 layers of LSTM with a RepeatVector layer in between has given good accuracy in training set. However, even after use of dropout layers and tuning based on number of neurons in each layer, the accuracy on test set is lesser than expected. Hence, model resulted in overfitting. The model can further improve with the introduction of more LSTM layers and optimizing the number of neurons. We got an accuracy of ~65% and BLEU score in the range of 0.12 to 0.37 for 4-grams.

Model 3: Design, train and test bidirectional RNN & LSTM model

It was observed that with lower number of epochs RMSprop fared pretty well and the effect of Adam was more visible as we increased the number of epochs. An early stopping criterion was used to converge and the model was able to achieve an accuracy of around 63%. Without treating some punctuations such as !, ?, . as separate words and lower-casing degraded accuracy and brought it down to the level of around 22%. We would investigate more on this front in the next phase. The BLEU score was in the range of 0.11 to 0.32 for 4-grams.

Model 4: Design, train and test Encoder-Decoder RNN & LSTM model

The validation accuracy is at 21% and does not change if we alter the Embedding size and no of LSTM units.

Finally, a prediction was attempted with some sample indices (from the Encoder input data). We have written a separate function to decode the sequences predicted by the model. This function takes the argmax of the prediction and locates the word sequence in the reverse output word indices.

# Model Tuning

## Model 1: Simple RNN LSTM Model

The model parameters are quite a few due to the inherent simplicity of the model. During the fit process, accuracy is used to measure model performance and for early stop. However, it is important to note that for language translation purposes, accuracy is not the most useful metric. The main reason for this is accuracy measurement tends to compare the actual and predicted values in an exact sense. Whereas the target values in a language translation problem are not exact values as in a typical classification problem. They tend to be more complex and at the same time, allowing some deviation without hurting the ultimate meaning of the sentence. **Due to this reason, it is decided that after the submission of interim report, the model tuning activity will continue using BLEU score. We have an initial version of using the BLEU score in the code which we will further work on as part of the final submission.**

In this phase, the following variables are varied as part of model tuning process:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Values Tried** | **Final Value Chosen** |
| NoOfNeurons | 256, 512, 1024 | 512 |
| Patience | 1, 2, 3, 4 | 4 |
| Epochs | 10, 20, 100, 200 | 100 |
| BatchSize | 16, 32, 64,128 | 16 |
| LearningRate | 0.01, 0.001, 0.0001 | 0.0001 |
| LearningOptimizer | Adam, RMS | RMS |

As mentioned above the translator model is preceded by a layer to embed words into a simple one hot encoding format with dimension equal to the size of vocabulary.

One only layer of LSTMs is used each of the encoding and decoding phases to maintain simplicity of the model. Translation using more sophisticated embeddings such as GLOVE and Bi-directional LSTMs is tried as part of subsequent steps in the project anyway.  
The highest accuracy (although accuracy is not the best metric for machine translations) achieved in this exercise is around 65%.

## Model 2: LSTM model with Embeddings

The initial LSTM model with embeddings had resulted in lower accuracy and high validation loss. Below actions were taken to tune the model:

* Use of GloVe embeddings had given wider coverage of words. The embedding layer was kept non-trainable initially but resulted in slow learning over a small batch size. The layer is made trainable now and shown good coverage of word embeddings over the small batch size.
* Use of 2 LSTM layers: The initial model had 1 LSTM layer with 256 neurons but model soon turned to be overfit from 2nd epoch itself. An additional LSTM layer is introduced along with 20% dropout to cater overfitting has resulted in better accuracy and machine translation. However, introduction of more LSTM layer remains an open topic for further improvements in machine translation modeling.
* Use of slow learning rate: The model seems to have performed well with RMSPROP optimizer with slow learning rate (0.001 to 0.0001). Also, feeding of sentences in small batch size through the model layers has given good translations and BLEU score for 1-to-4-grams for the whole training and testing sets.
* Highest accuracy of 65% is observed on training set of size 4000 records.

Model 3: Design, train and test bidirectional RNN & LSTM model

Various values of learning rate, batch size, and number of neurons were tried along with the change in optimizer. The model was seen to perform better with lower batch sizes. This also reduced the problem of overfitting that was observed with larger batch sizes. Also, increasing the number of epochs to 100 resulted in the max test accuracy beyond which the training accuracy improved but the test accuracy remained stable resulting in overfitting. The test accuracy at 1000 epochs was only slightly higher than at 100 epochs which in turn was about 2% higher than at 30 epochs. It was decided to keep the RMSprop optimizer with Early Stopping at lower epochs to get the best outcome.

Model 4: Design, train and test Encoder-Decoder RNN & LSTM model

Tuning was not attempted for this model.

# Performance of Best Model

Model 2: LSTM model with Embeddings has been so far the best performing model with the highest accuracy of 65% and also a better BLEU score on average for 4-grams.

# Insights Gathered

* + 1. While the language translation typically uses the target values to backpropagate and fine tune the learning just as in a typical classification problem, it is very different in terms of expectation regarding the outcome.   
         
       Typically, in a classification problem the target actual value is compared with the target predicted value in an exact sense.   
         
       However, when it comes to language translation, an overall meaning should be compared rather than the exact sentences. This renders the most used metric, ‘accuracy’, useless.   
         
       In this context, various metrics used for measuring linguistic similarity and correctness of the predicted sentence by the contemporary research community are studied by the project team. For simplicity, it is decided that BLEU score will be used in this project context.
    2. Stop words are typically removed in a classification problem such as sentiment analysis. However, removing stop words may leave the target translated sentences meaningless for humans. Hence the project team has decided to keep the stop words.  
         
       As a consequence, because the number of stop words occur more frequently in any language, their sample is overwhelmingly higher compared to other words.   
         
       This problem seemed to have exaggerated especially due to the fact that only sentences of length smaller than eight are used for training and testing.   
         
       Since the data sample used is small and the networks used are also simplistic due to the limited computational power available, the predicted or translated sentences seemed to have more stop words in place of actual words.   
         
       It Is felt that the learning could have been better, if the networks are more complex with higher number of parameters and the data sample used is also sufficiently larger.
    3. Bidirectional LSTM seemed to have performed significantly better in alignment with common intuition.   
         
       The team felt the main contributor for the difference is the capability of Bidirectional LSTM to learn context of a word.   
         
       It is observed that when simple uni-directional LSTM is used, the translated sentences contained too many stop words most probably due to their significantly higher frequency, jeopardizing the accuracy.   
         
       Since Bidirectional LSTM is capable of learning context of a word due to its reverse parsing, it seemed to have overcome the disadvantage of imbalanced sample favoring stop words.
    4. Unlike other problems in machine learning, the expected target values are exact in the sense either they match or do not match.   
         
       When it comes to language translation, even by humans, the target sentences tend to vary while preserving the meaning.   
         
       The variation is not only in the choice of vocabulary but could also be in the way sentences are constructed such as active voice or passive voice, the order of subjects and predicates, choice of analogies peculiar to the cultural differences, and many more.  
         
       For hypothetical perfect learning, the model should have not only sufficiently large number of parameters but also the training dataset should have all variations in sentences sufficiently balanced in their sample sizes. Besides this it should be a given fact that the translation quality is also good enough.  
         
       In reality, it is observed that even verifying the training datasets from the above perspective is very challenging.   
         
       It seems obvious that the current state of the art Machine Translation models consists of not only highly complex structures with very high number of learnable parameters but are also trained on humongous data requiring monumental computational resources possible only by large corporations.
    5. The models initially had a lower accuracy of around 20% to 25% when no punctuations which come at the end of a sentence were not treated separately as words. When the punctuations !, ? and . were treated separately as words and also lowercasing was not done through the tokenizer, the accuracy improved quite a bit and became around 60% to 65%. This helped the models to understand the context of a word in a more meaningful way and thereby was able to aid in assigning the right amount of importance to different words for proper classification.
    6. As shown in the html file submitted with this report, the BLEU score came out to be the best for the RNN + LSTM model with embeddings. It was slightly better than the bidirectional LSTM model but fared much better than the single RNN + LSTM model.

# Next Steps

After submission of interim report,

* A different data sample with varying line lengths will be used.
* Further tuning of the models will be performed.
* User interface will be developed to translate English sentences into German using the best performing model.
* Check if we need to apply any additional pre-processing to the dataset.
* Try different optimizers and alter the learning rate.

# Final Thought

We were expecting the encoder/decoder model to give the best results which is not the case here. This might be because we did not use any attention layer. We will investigate further in this direction.