NEURAL MACHINE TRANSLATION

FROM NAHUATL TO SPANISH

Content

Introduction

This project aims to create neural machine translation from the Uto-Aztecan language Nahuatl to Spanish. To achieve this, a Seq2Seq model with encoder and decoder LSTMs has been trained on 17071 parallel sentences in Nahuatl and Spanish. The model predicts the Spanish translation of a Nahuan sentence word by word based on an encoded representation of the source sentence and a probability distribution over the Spanish vocabulary. The sparse and sometimes faulty data in combination with a perhaps too simple model architecture makes this a difficult task. After experimenting with hyperparameters and the training data, the model reaches a BLEU score of 0.05 on unseen test data, which is very low.

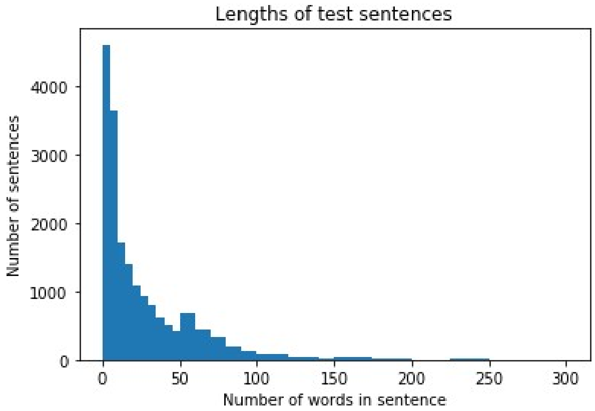
Data

Training models on large sets of parallel data is a successful method for neural machine translation. However, most of the world's languages lack the big parallel corpora necessary for training such models. Creating neural machine translation for low-resource languages like Nahuatl is therefore a difficult task. Fortunately, there is a corpus of parallel sentences in Nahuatl and Spanish consisting of almost 18 thousand sentences. While this may not be enough to obtain excellent performance, it is a good start.

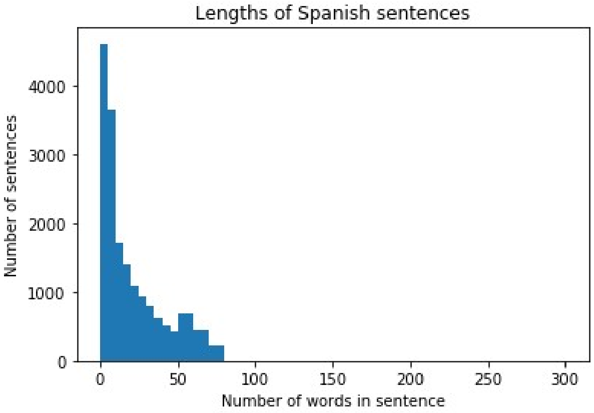
The parallel corpus

The parallel sentences used to train the model are retrieved from the Axolotl Corpus (GIL UNAM, 2015). It is the largest parallel corpus of Nahuatl and Spanish. The data comes from Nahuan books and their Spanish translations which have been scanned and processed. The most represented genres in the corpus are historical, short stories and literature and it also contains texts of the types didactic, recipes, magazines, musical and legal. 45.7% of the data is text in Classical Nahuatl and 54.3% is text in several modern varieties (Gutierrez-Vasques et al., 2016).

The length of the sentences in the parallel corpus varies. 1667 sentences consist of just one word in their Spanish versions and seem to be titles. At the same time, the longest sentence in the data is 3664 words long in its Spanish version. Most sentences are shorter than 50 words, but a sentence of 50 words is very long and 2077 of the Spanish sentences in the data are longer than that. 372 sentences contain more than 100 words, and 64 sentences contain more than 200 words.



An examination of the longer sentences shows that they often contain several sentences and, in a few cases, whole articles. The data processing appears to have failed since these sentences lack punctuation to mark the sentence borders. Instead, only capitalized letters mark what should have been a sentence boundary. This is unfortunate, since correctly processed data would have resulted in more parallel sentences to train on. It also negatively affects how the model handles word order when the sentences the model trains on are not separated properly.



Because of the incorrectly separated text that result in some extremely long sentences, 824 sentence pairs with Spanish versions of more than 75 words were removed from the dataset. There are still faulty sentences in the data consisting of several short ones but examining all of them and manually separating the incorrect ones would be too time consuming. By removing the 824 longest ones the performance of the model improves despite having less data to train on. The BLEU score increases from 0.01 to 0.04 with the longest sentences removed from the training and testing data.

While the parallel corpus is of a considerable size, it is very diverse and has some serious flaws. It contains Nahuan text from different centuries, genres, and varieties and even within the varieties, the authors make use of different spelling norms. It is also problematic that the most represented variety is the historical and now extinct Classical Nahuatl. Besides, many parallel sentences are in fact several sentences that have been incorrectly separated. In addition to this, a large share of the parallel sentences (12%) consists of single words that do not let the model learn word order or which words tend to cooccur in a sentence. In the one-word sentences, Nahuan words sometimes occur on the Spanish side, due to faulty processing of the books, perhaps the didactic material. These shortcomings together probably have a negative impact on the performance of the model.

Pre-processing the data

The first step of the pre-processing of the data is to remove punctuation from the sentences and lowercase all tokens. The apostrophes in the Spanish words are not removed since they often distinguish between homonyms. E.g., the word 'tú' means 'you' while 'tu' means 'yours' and the word 'está' means 'it is' while 'esta' means 'this'. In order for the model to produce well-formed Spanish sentences it needs to handle these apostrophes correctly and this is the reason the apostrophes are not removed with the rest of the punctuation.

Regarding the Nahuan sentences, there is a greater variation in spelling both between and within the varieties. The Python package Elotl (Aguilar & Pugh, 2021) provides spelling normalization for Nahuatl which is applied to the Nahuan data. The package offers normalization according to three different spelling norms and for this project, the spelling norm INALI is chosen since it is used by Instituto Nacional de Lenguas Indígenas and common in research on Nahuatl. According to this spelling norm, the graphemes <w>, <k> and <h> represent the phonemes /w/, /k/ and /h/. With the Nahuan sentences normalized according to the INALI spelling norm, the BLEU score is slightly higher than for the other two spelling norms.

The model trains on a Spanish vocabulary of 10451 words and a Nahuan vocabulary of 12845 words. All words that occur at least twice in the dataset are included in the vocabularies. When even the words that only occur once are included in the vocabulary, their size increase to 27313 Spanish words and 49302 Nahuan words. Since Nahuatl is an agglutinative language and Spanish is not, the Nahuan vocabulary is considerably larger. For this reason, limiting the vocabulary might have greater consequences for Nahuatl than for Spanish. While limiting the vocabularies is not optimal, especially not the Nahuan one, it is necessary to constrain them for efficiency since SoftMax is calculated over the entire target vocabulary in the loss function.

Each sentence is given a start token and an end token to mark the sentence boundaries. The individual words in each vocabulary, as well as the start- and end tokens, are given a unique number. The sentences are encoded with the corresponding indices of the tokens before they are given to the model. The data is split into training, validation, and test sets of the sizes 16 000, 500 and 571. The subsets are further divided into batches of size 64.

Model

The model is Seq2Seq architecture with an LSTM encoder and a LSTM decoder. The Seq2Seq model gives the Nahuan source sentence to the encoder which creates a representation of it. The hidden state of the encoder is passed to the decoder which maps this representation to words in the Spanish vocabulary. The models have been designed with inspiration from code on Ben Trevett's GitHub (2018) which in turn is based on the paper Sequence to Sequence Learning with Neural Networks (Sutskever et al., 2014).

Encoder

The task of the encoder is to create an abstract representation of the Nahuan source sentence. The encoder takes a tensor with the size of the Nahuan vocabulary representing the sentences in the batch and creates an embedding of size 256. When training the model, dropout of 0.5 is applied to this embedding. The embedding is then given as input to an LSTM with two layers and a hidden size of 1024. The output of this LSTM is not of interest, but the hidden state and the cell state are what is passed on to the decoder.

Decoder

The task of the decoder is to map the abstract representation of the Nahuan sentence from the encoder to a Spanish sentence. The hidden state and the cell state of the encoder is given as input to the decoder as well as a tensor with the start-token of the Spanish target sentence which the decoder embeds. The embedded tensor, together with the hidden and cell state from the encoder is given as input to the decoder LSTM. A linear layer is applied to the output of the decoder LSTM which maps it to the Spanish target vocabulary.

Seq2Seq

The task of the Seq2Seq model is to organize the encoder and decoder properly and predict the Spanish translation of the Nahuan sentence. First, the source sentence is given to the encoder which returns the hidden and cell state representing it. Then, the Seq2Seq model iterates over the target sentence and on the first iteration the start-token of the target-sentence and the hidden and cell from the source sentence are given as input to the decoder. In other words, the decoder predicts what token follows the start token of the Spanish sentence, given the abstract representation of the Nahuan source sentence.

The iteration continues and the next prediction is based on the output and the hidden and cell state of the previous prediction. To help the model learn faster, teacher forcing is applied during training. The ratio is set to 0.5 which means that in 50% of the cases, the model's prediction is used to predict the following token. But the other 50% of the time, the ground truth is chosen instead, and the decoder makes its prediction based on that. This is a method to ensure that the model will learn what the correct output looks like and this way speed up and improve the learning.

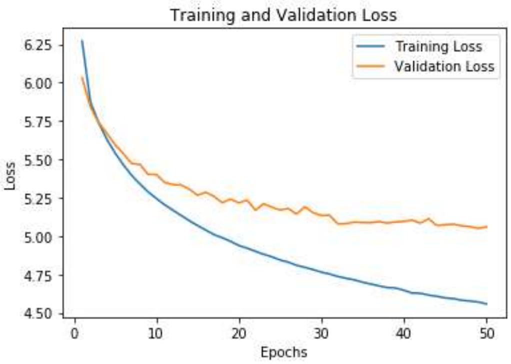
Training

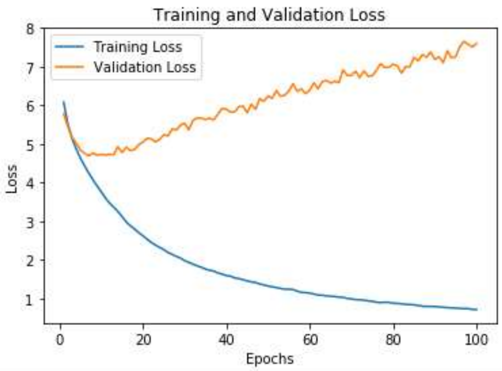
The loss function used is Cross-Entropy Loss since it calculates SoftMax on the output of the Seq2Seq model. Both training and validation loss are implemented and calculated in the training loop and visualized in a diagram to better understand the training progress. While the training loss decreases with each epoch, the validation starts to increase after some iterations, depending on the hyperparameters. The aim is to stop the training when the validation loss starts to increase but continue as long as it decreases to prevent over- and underfitting. However, there does not seem to be a direct correlation between a low validation loss and a high BLEU score (see Testing and Evaluation).

Hyperparameters

The most successful model trained for 30 epochs with a batch size of 64, embedding size of 256, hidden size of 1024, learning rate of 0.001, two layers and dropout of 0.5. While the models trained with other hyperparameters do not perform as well, experimenting with them led to interesting discoveries. A model with smaller embedding and hidden size does not overfit but also fails to learn properly. The curve representing the validation loss of this model flattens out but never reaches a value below 5. The model with larger embedding and hidden size on the other side manages to learn so well that it eventually starts to overfit on the training data at the expense of the validation loss.

The diagrams below show the training and validation loss of two models with very different hyperparameters. The validation loss of the model with lower embedding and hidden dimensions flattens without increasing trained on 50 epochs. The validation loss of the more successful model however increases after around 10 epochs. The left model is trained with a batch size of 8, embedding size of 32 and hidden size of 32. The model to the right is trained with a batch size of 64, embedding size of 256 and hidden size of 1024.





After observing the training and validation loss of the more successful model to the right, it seems reasonable to train the model for around 10 epochs. Surprisingly, a model trained for 10 epochs is not the most successful. Despite the fact that the validation loss increases after 10 epochs, the Spanish sentences generated by the model seem to be more grammatical if the model trains for more epochs. The obtained BLEU score is also higher compared to a model that trains for only 10 epochs. In the following section, this will be discussed further.

Testing and evaluation

The evaluation metric BLEU score is used to test the performance of the model. It is a score between 0 and 1 that indicates how similar a translation is to another where 1 is assigned to two identical sentences. BLEU is an automated method approximating human evaluation of translations (Papineni et al., 2002). In this case, the Spanish translation of a Nahuan sentence produced by the model is compared to the Spanish target sentence from the parallel corpus. While we want the BLEU score to be as high as possible, it is not necessary to obtain a score of 1. This is because human translations would not be identical either and we want the neural machine translation to resemble human translation.

Sentence lengths	**BLEU score**
1-3 words	0.22
3-5 words	0.08
6-9 words	0.04
11-19 words	0.07
23-38 words	0.03
43-68 words	0.05

When the model is evaluated, it translates the 571 test sentences and obtains a BLEU score of 0.05. This is very low and indicates that the model fails to translate properly. An observation made is that the BLEU score seems to be dependent on the length of the translations that the model generates. The table to the right displays how different the model performs on sentences with different lengths. The performance on the shortest sentences is decent while the model struggles with the very long sentences. As mentioned above, many sentence pairs in the data are very short or consist of only one word and the model is therefore used to handle them from the training. Likewise, many of the very long sentences are not split correctly which causes the model to learn incorrect word order. This is probably why the longer sentences receive such a low BLEU score.

It is hard to say what the relationship between validation loss and BLEU score looks like. As mentioned above, the highest BLEU score is not obtained by the model with the lowest validation loss. In the table below, the BLEU score for different models that have been trained for different numbers of epochs and with different validation losses are presented. It clearly shows how a lower validation loss does not guarantee a higher BLEU score.

Epochs	10	20	30	100
Training loss	3.77	2.57	2.00	0.72
Validation loss	4.69	5.11	5.41	7.6
BLEU score	**0.03**	**0.05**	**0.05**	**0.05**

Another way to evaluate the model's performance is to look at the sentences it produces. The four models from the table above produces four different translations of the Nahuan sentence *nonantsin onoitstilok* which means *my mother has gotten a cold*. The Spanish translation in the corpus is *mi mamá se ha resfriado*. All other hyperparameters are the same, only the number of epochs and the final validation loss differ. For all four models, this sentence was in the test set and had not been seen by the models during training.

While it is not enough to evaluate the model's performance on just one test sentence, the translated sentences in the table give an idea of how the number of epochs and validation loss affect it. Especially since the model's performance varies dramatically between long and short sentences, the results may look very different for sentences of different lengths. However, by examining the grammaticality of the generated sentences one can see that there are some obvious errors.

10 epochs	mi mamá se
20 epochs	mi mamá te ha resfriado
30 epochs	mi mamá te lo ha dicho dicho
100 epochs	mi mamá te ha ha dicho

After 10 epochs, the model outputs an incomplete sentence. It chooses the correct reflexive pronoun but omits the verb and is therefore not a valid sentence. After 20 epochs, the model outputs a sentence that is very close to the target sentence – only the reflexive pronoun is wrong and ungrammatical. After 30 and 100 epochs we can assume that the model has overfitted on the training data and we no longer generates the word for cold. There is also ungrammatical repetition of words in the last two sentences. The best sentence is without doubt the one produced after 20 epochs although the BLEU score was slightly higher after 30 epochs. These observations show that even though the validation loss is lower after 10 epochs, it does not mean that the model performs better.

Analysis and Discussion

The evaluation of the model and the observations made raise some questions. The code used as inspiration for this translation project trained a similar model on 30 thousand parallel sentences in German and English and obtained a BLEU score of 0.4. This is quite good, especially compared to the very low score of 0.05 for this model. Why does this model perform so badly? There are a few obvious differences between the German-English translation model and the one for Nahuan-Spanish.

Firstly, German and English are similar languages from the same language family and the Seq2Seq approach is obviously successful for translating between the two languages. While Nahuatl and Spanish have a long history of language contact and contain several loan words from each other, they are structurally very different. As mentioned above, Nahuatl is agglutinative, and Spanish is not. This means that what Nahuatl can express with just one word requires several words to be expressed in Spanish. The model architecture chosen for this project requires longer Nahuan words consisting of several meaningful units to be treated as single words. Limiting the size of the vocabulary will have consequences for a language like Nahuatl, since there are twice as many unique Nahuan words in the parallel corpus than there are Spanish unique words.

Secondly, the dataset used in the English-German example is almost twice as big as the parallel corpus in Nahuatl and Spanish which will have an impact on the performance of the models as well. Sparse parallel data is the main challenge when it comes to neural machine translation for low-resource languages. Nevertheless, the quality of the parallel data also matters. The Nahuan sentences used to train the model come from different centuries and varieties and will therefore be hard for the model to generalize. Besides, many sentences are not separated correctly. Due to the flaws discovered in the Nahuan-Spanish parallel corpus and its great diversity, one can suspect that the German-English dataset is not only bigger but also better and more uniform which can explain the higher performance.

However, the low BLEU score obtained does not mean that the model is bad at translating from Nahuatl to Spanish. For many test sentences, the model seems able to map the Nahuan sentences to equivalent sentences in Spanish since many words and topics are correctly translated. The main issue is the grammaticality of the generated Spanish sentences and probably the primary reason the BLEU score is so low. In other words, the model is generally bad at generating grammatical Spanish sentences and not necessarily bad at mapping meaning in Nahuatl to meaning in Spanish.

Since more parallel data in Spanish and Nahuatl is not easily available, one way to improve the performance of the model could be to train it on more sentences in Spanish. When the grammaticality of the generated sentences increases, the BLEU score would probably increase too and hopefully the usefulness of the model. Further improvements could be made by implementing a bidirectional LSTM, since many sentences in the data are long, or attention.

Conclusion

This project attempts neural machine translation from the low-resource language Nahuatl to Spanish using a Seq2Seq architecture. While the model to some extent succeeds in mapping Nahuan sentences to Spanish sentences, the grammaticality of the generated sequences is poor which results in a very low BLEU score of 0.05. Structural differences between the languages seem to be a reason for the weak performance, as well as very diverse and sometimes faulty parallel data. While the need for large parallel data is obvious, it is also clear that the quality matters. The performance could improve dramatically by fixing the faulty sentences and train the model on monolingual data in Spanish. Implementing bidirectional LSTMs and attention might also improve the model's performance.

References

Aguilar, P. & Pugh, R. (2021) *Py-Elotl.* [Python package]. https://pypi.org/project/elotl/

Grupo de Ingeniería Lingüística GIL, UNAM. (2015). *Corpus paralelo español-náhuatl.* [Dataset]. http://www.corpus.unam.mx/axolotl

Gutierrez-Vasques, X., Sierra, G.E. & Pompa, I.H. (2016). Axolotl: a Web Accessible Parallel Corpus for Spanish-Nahuatl. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16).* 4210–4214. https://aclanthology.org/L16-1666

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics.* 311–318. 10.1.1.19.9416 https://aclanthology.org/P02-1040.pdf

Sutskever, I., Vinyals, O. & Le, Q.V. (2014). Sequence to Sequence Learning with Neural Networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems*, MIT Press, MA. 3104-3112. https://arxiv.org/pdf/1409.3215.pdf

Trevett, B. (2018). *Sequence to Sequence Learning with Neural Networks.* [Jupyter Notebook]. https://github.com/bentrevett/pytorch-seq2seq/blob/master/1%20-%20Sequence%20to%20Sequence%20Learning%20with%20Neural%20Networks.ipynb