Europarl Corpus and Machine Translation

David Barbas Rebollo

Automatic Translation
February, 2021

1 Introduction

This project conducts the exercises proposed for the final lab assignment of this subject. These include doing experimentation to find the best statistical machine translator via Moses[3], finding the best neural machine translator via NMT-Keras[5], experimenting with the Transformer model provided by NMT-Keras and finally evaluating a different neural toolkit. OpenNMT[1] toolkit was selected for this last task.

Moreover, the dataset used is a subset of the Europarl[2] dataset which contains 50,000 training samples and 1000 test samples. This parallel corpus is extracted from the proceedings of the European Parliament. It includes 21 different languages but this work focuses only in translation from English to Spanish.

The objective proposed in the lab assignment is to obtain models able to achieve a value of BLEU[4] around 25.5.

2 Experimental Evaluation

This section describes the different experiments conducted and the why they were chosen. As well as, the metrics used to compare the models.

2.1 Evaluation Metrics

The metric used for the evaluation of the models is the Bi-Lingual Evaluation Understudy, BLEU, which evaluates the quality of a translated text.

2.2 Preparation of the environment

To conduct the different experiments the EVIR environment offered by the university was considered unfeasible for the experimentation, due to the extremely high temporal cost involved. Therefore, all experiments where done locally using docker. The hardware used was:

- Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
- NVIDIA GeForce MX150 (2GB GDDR5)

2.3 Data preparation

Firstly, the Europarl subset data is downloaded. Consisting of 4 files, 2 for training and 2 for testing in English and Spanish. Then the training data was split to form a training and validation of 47,000 and 3,000 respectively.

Secondly, all sets were cleaned and tokenize with X for Moses' and NMT-Keras' experiments. Meanwhile, for the Open-NMT experimentation their toolkit preprocess function was used.

It is important to note that all sequences length were limited to 60.

2.4 Basic Exercises

This section includes the description of the basic exercises conducted in this project.

2.4.1 Moses Experiments

Given the insights of the previous assignment related to statistical machine translation and that the increased computation time required, the parameters experimented were reduced. Only the most relevant ones were considered. Also, by this means it is possible to compare if these parameters still offer good results with a much bigger dataset.

The experiments involve using different size n-grams, the weights adjustment method, the discount method and the number of iterations.

2.4.2 NMT-Keras Experiments

Likewise, for the experimentation related to NMT-Keras these were limited and selected scrupulously. Therefore, given the insights from the neural machine translation assignment the experimentation involved only the most relevant parameters, word embedding and encoder-decoder size. This way, again, it is possible to compare if these parameters are still the most relevant to obtain a good performance with a bigger dataset. Additionally, training duration was fixed to 5 epochs.

Furthermore, for all neural translation models only the adam optimizer was only considered as it offers faster convergence than adadelta and adragad.

It is also very important to notice that as the model complexity increased the batch size needed to be reduced by the memory limitation of the GPU used. Goes without saying, this could counterproductive for training as it harmed batch normalization.

3 Advanced Exercises

This section includes the description of the advanced exercises conducted in this project.

3.1 NMT-Keras's Transformer

For this exercise, again experimentation started with the best performing transformer models in the previous lab assignment with the EuTrans dataset.

Given this starting point, as with the attention encoder decoder models only the embedding and encoder-decoder size parameters were varied. However, due to the hardware limitations and the complexity of these type of models, experimentation was extremely limited. This is reflected in the experimentation section, were it is visible the exploration of these parameters. Additionally, training duration was fixed to 5 epochs.

3.2 Open-NMT Toolkit Evaluation

The last experiment involves the use of of the Open-NMT toolkit. It involves trying to obtain similar results to the one obtained with NMT-Keras. The experimental procedure was very similar as to the one done with NMT-Keras as both offer the implementation of attention encoder decoder models.

The biggest difference between the both is how to execute experiments. The reason of this statement is that unlike NMT-Keras that a specific configuration file needs to be written explicitly, Open-NMT enables users to completely define the structure of the network which the user wants to train via terminal. This becomes very useful to script a pipeline for several models.

To do a fair comparison of both toolkits, the best 2 performing models were selected to replicate their configuration in Open-NMT. Additionally, training duration was fixed to the number of time steps equivalent to 5 epochs.

4 Results

4.1 Basic Exercises

This section exposes the results obtained by the basic exercises accompanied by the insights obtained by them.

4.1.1 Moses Experiments

Table 1: Results comparison of Moses experiments.

Weights Adj.	Max It.	N-grams	Discount	BLEU
MERT	5	3-grams	Kneser-ney	25.29
MERT	5	4-grams	Kneser-ney	24.67
MERT	5	5-grams	Kneser-ney	26.49
MIRA	5	5-grams	Kneser-ney	26.48
MERT	5	5-grams	Witten Bell	25.68
MERT	5	5-grams	Good Turing	25.66
MERT	15	5-grams	Kneser-ney	26.13

The first parameter explored was the number of grams used. As for this task 5-grams showed the best performance, it was also used for the rest of the experimentation.

The adjusting of weights method was considered non-relevant for the improvement of the model's performance. So Mert was maintained as it offered better results in previous assignments.

In addition, the weight adjustment converges fine with maximum 5 iterations and increasing this parameter does not offer better results.

Finally, despite not varying much, Kneser-ney seems to be the most appropriate discount method.

Therefore, given the results obtained from these experiments show that for this task the third experiment offers the best parameters for a statistical machine translator.

4.1.2 NMT-Keras Experiments

Table 2: Results comparison of NMT-Keras experiments using the attention encoder decoder model.

Model Type	Batch Size	Emb. Size	Model size	Optimizer	BLEU
Enc-Dec	16	128	128-128	Adam	16.83
Enc-Dec	12	256	64-64	Adam	18.92
Enc-Dec	8	256	128-128	Adam	18.67
Enc-Dec	4	512	128-128	Adam	12.04
Enc-Dec	4	512	256-256	Adam	9.86

Firstly, is important to notice how the batch size decreased with increasing model complexity. Despite using a small batch size by default decreasing it more harms the batch normalization used to increase performance. This and the low number of epochs are most likely the poor performance of more complex models. As there were too many parameters to learn in very few epochs. these comments

Although more complex models should yield a better performance, from the results obtained, it is possible to observed that it is important to not build an overcomplex model. Since it is not available to learn correctly the patterns of the training data in a small period of training.

Also, it is also visible that the embedding size is more important that the model size.

Therefore, the parameter balance of experiment 2 offers the best performance of the experimentation as in the previous lab assignment.

4.2 Advanced Exercises

This section exposes the results obtained by the advanced exercises accompanied by the insights obtained by them.

4.2.1 NMT-Keras's Transformer

Table 3: Results comparison of NMT-Keras experiments using the transformer model.

Model Type	Batch Size	Emb. Size	Model size	Optimizer	BLEU
Transformer	16	128	128-128	Adam	0.22
Transformer	8	256	256-256	Adam	0.17

The experimentation of this type of neural machine translation model was even more limited and results were even worse. The trained models are not capable of learning the features of the training set properly leading to a BLEU-score under 1.0.

A solution to this problem could be to use a increased batch size as well as increasing the number of epochs to ensure the proper training of the model. As this type of model gives state-of-the-art results in machine translation problems.

4.2.2 Open-NMT Toolkit Evaluation

Table 4: Results comparison of Open-NMT toolkit experiments.

Model Type	Batch Size	Emb. Size	Model size	Optimizer	BLEU
Enc-Dec	16	128	128-128	Adam	16.27
Enc-Dec	12	256	64-64	Adam	17.88

For the last experiments, using Open-NMT toolkit, it was key to replicate as accurately as possible the same parameters used in the NMT-Keras attention encoder decoder model.

It can be seen that results obtain are very similar. As the experiment models yield a BLEU score which varies less than 1 from the original models. Given this circumstance, although not obtained outstanding results, it is possible to conclude that the toolkit was used correctly for this task.

5 Conclusions

Completing this project has been really useful and interesting because of different factors.

In first place, by using a dataset with an huge increase in samples, compared to the other lab assignments, it is possible to really understand the temporal complexity of machine translation. Meaning it is of high importance to select correctly the parameter configuration of the machine translation model.

In second place, the use of docker has eased the use of these toolkits in an incredible manner. Which has accelerated the experimentation once docker was fully mastered.

Additionally, is it important to notice that the default resources for student to conduct experimentation was very lacking. Therefore, different students were able to explore more than others in experimentation. So, it is considered that the complexity of the task proposed exceeds the available resources.

Furthermore, despite the limited experimentation done with Open-NMT it considered as a better toolkit than NMT-Keras since its configuration method can be classified as superior. However, a huge inconvenience is that the training duration can not be set as epochs and must be set as time steps. Therefore, it was tedious to calculate the number of time steps to perform a fair evaluation of the toolkit.

To conclude, comparing the results obtained during the experimentation the performance of statistical translators was much greater than the performance offered by the neural models. In practice, neural translators should obtain similar or higher results so it is most likely that the use of small batch size and few epochs are the cause of this poor performance.

References

[1] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[2] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

[3] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, page 177–180, USA, 2007. Association for Computational Linguistics.

[4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[5] Álvaro Peris and Francisco Casacuberta. Nmt-keras: a very flexible toolkit with a focus on interactive NMT and online learning. *CoRR*, abs/1807.03096, 2018.