Alexandra Butler
Cameron Duval

NMT Training and Morphological Learning of Low-Resource Languages

Introduction

Neural machine translation has been proven a highly effective tool for interlingual translation and has thus only increased in popularity over the past decade. While these models are very powerful and only use a fraction of the memory required for statistical machine translation models (Cho et al., 2014), they still come with their caveats. The largest of these caveats is the truism that these models can only run accurately if they are given enough text, which could then make the accessibility of these models more difficult for low-resource languages. Hupa (Pacific Athabaskan) and Croatian (Slavic) are both languages that do not have as many linguistic resources for online translation as their genealogical and geographic neighbors, namely Russian and English. This project will investigate the training of two separate neural models for Hupa-English and Croatian-English translation. Along with general translation and training accuracy, we will use our models to assess to what extent they are learning the morphology of our respective languages, more specifically noun declension in Croatian and verbal morphology in Hupa.

Background

Previous Research

OpenNMT is an open-source toolkit designed to support work on machine translation, as well as for tasks related to natural language processing such as summarization and parsing. Released in 2016, it was created using standard sequence-to-sequence models in addition to attention-based ones. Klein et al. (2018) assessed its translation accuracy with a German-English parallel corpus. The paper found that neural machine translation is an improvement on statistical machine translation in terms of accuracy, model size, and by extension, efficiency. The accuracy of NMT for large world languages is generally high. Though the authors did not report the desired BLEU (a number between 0 and 1 which is used to evaluate MT), the score of about 0.25 was comparable to and in some experiments better than most of the other state-of-the-art systems used for MT at the time. The experiment involved the use of an encoder-decoder architecture and Byte-Pair Encoding tokenizer. Two years later, some of the same authors recreated their experiments having switched to a different transformer architecture and the SentencePiece tokenizer which was the same one used for this paper. The BLEU score reported in the 2020 paper was about 0.41, much higher than in the first round of experiments.

Research Questions

Building on the work of Klein et al. (2018, 2020) which examined MT for German, a historically widely-spoken and well-documented language, the aim of this paper was to apply some of the same methods to low-resource languages and test the translation accuracy. The questions posed at the beginning of this project included the following:

a) How accurately can an OpenNMT sequence-to-sequence language model translate low-resource languages, specifically Hupa and Croatian?
b) How does the model's translation accuracy compare to previous attempts at translation for bigger WLs?
c) How well does the model capture morphologically complex structures (conjugation, declension)?

The last question was not answered within the scope of this project but it could be a next step.

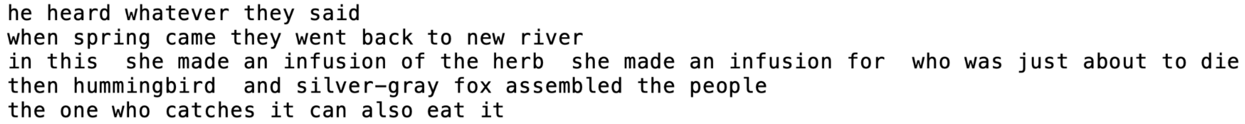
Data and methods

Model and Interface

For this project, we used the OpenNMT.py model, a PyTorch version of the open-source neural machine translation model, OpenNMT, which can be easily accessed and trained from any data given to it. OpenNMT is a sequence-to-sequence recurrent neural network with a default of two hidden layers. OpenNMT starts with constructing word vectors of the source language, then using these vectors in the next sequence to calculate the probability of the target language translation. With each word in the target language, the predicted probability of the previous word is used as input to the hidden layer of the next word (Klein et al., 2018). We used Google Colaboratory to run OpenNMT, as it is a convenient, web-based interface to run Python on a GPU which heavily improved our processing times.

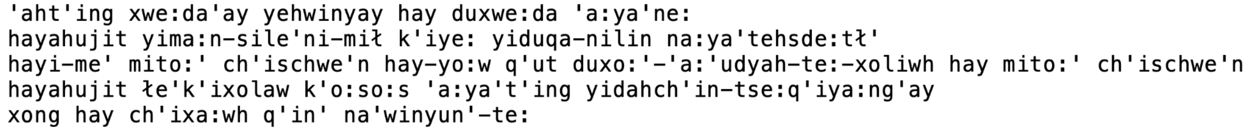
Corpora

The Hupa corpus is comprised of a collection of pre-translated narratives. The total line count of the parallel translations is 6765 and it is comprised of ~82,000 English tokens (8,789 types) and ~46,000 Hupa tokens (11,955 types). Some preprocessing was done initially to help clear out potential errors, including removing punctuations and capitalizations, removing double translations on the same line, and randomizing the line order to avoid the model training only on vocabulary from certain narratives. Training, validation, and test corpora were made from the original by separating them 80%, 10%, and 10% respectively.



```
he heard whatever they said
when spring came they went back to new river
in this  she made an infusion of the herb  she made an infusion for  who was just about to die
then hummingbird  and silver-gray fox assembled the people
the one who catches it can also eat it
```

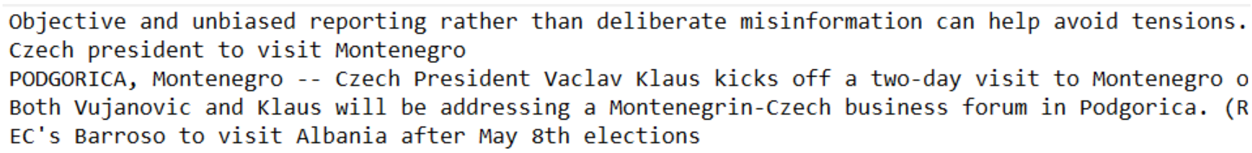
English training set



```
'aht'ing xwe:da'ay yehwinyay hay duxwe:da 'a:ya'ne:
hayahujit yima:n-sile'ni-mił k'iye: yiduqa-nilin na:ya'tehsde:tł'
hayi-me' mito:' ch'ischwe'n hay-yo:w q'ut duxo:'-'a:'udyah-te:-xoliwh hay mito:' ch'ischwe'n
hayahujit łe'k'ixolaw k'o:so:s 'a:ya't'ing yidahch'in-tse:q'iya:ng'ay
xong hay ch'ixa:wh q'in' na'winyun'-te:
```

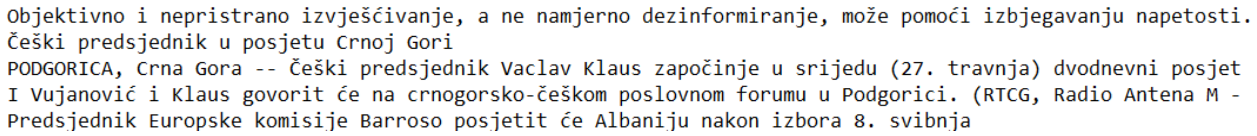
Hupa training set

193,388 parallel lines were selected from the Croatian-English corpus "SETimes" for the purposes of this project. The data mainly consists of news articles and Wikipedia entries. The Croatian data is made up of 39,976 tokens and the English data of 26,400 tokens. As with the Hupa-English corpus, the Croatian-English parallel lines were split three ways: 80% for the training set, 10% for validation, and 10% for the test set. No punctuation was removed during the preprocessing stage as punctuation conventions differ greatly in standard Croatian and English and removing it may impact the assessment of the translation.



```
Objective and unbiased reporting rather than deliberate misinformation can help avoid tensions.
Czech president to visit Montenegro
PODGORICA, Montenegro -- Czech President Vaclav Klaus kicks off a two-day visit to Montenegro o
Both Vujanovic and Klaus will be addressing a Montenegrin-Czech business forum in Podgorica. (R
EC's Barroso to visit Albania after May 8th elections
```

English training set



```
Objektivno i nepristrano izvješćivanje, a ne namjerno dezinformiranje, može pomoći izbjegavanju napetosti.
Češki predsjednik u posjetu Crnoj Gori
PODGORICA, Crna Gora -- Češki predsjednik Vaclav Klaus započinje u srijedu (27. travnja) dvodnevni posjet
I Vujanović i Klaus govorit će na crnogorsko-češkom poslovnom forumu u Podgorici. (RTCG, Radio Antena M -
Predsjednik Europske komisije Barroso posjetit će Albaniju nakon izbora 8. svibnja
```

Croatian training set

Methods

For both Croatian and Hupa, our methods ran the same in terms of training and testing the model. Once we have uploaded our corpora into Colab, the process involved installing OpenNMT, changing the configuration file to direct to our data (as well as altering aspects of training such as steps and dropout), building vocabulary, training the model, then eventually using the saved model to translate our test corpora and assess translation accuracy. BLEU score was also calculated for the predicted translations using Colab as well. All graphs and data regarding training and validation accuracy were inputted into CSV files and visualized using Python's Pyplot tool.

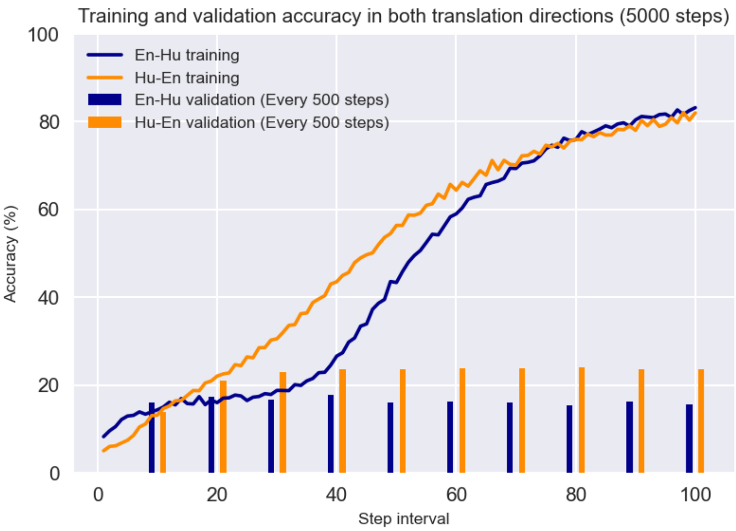
Partway through the project, we also started using the SentencePiece tokenizer on our corpora to tokenize our data at a sub-lexical level. This was in the hopes of improving translation accuracy as well as better learning the morphological aspects of both languages.

Results

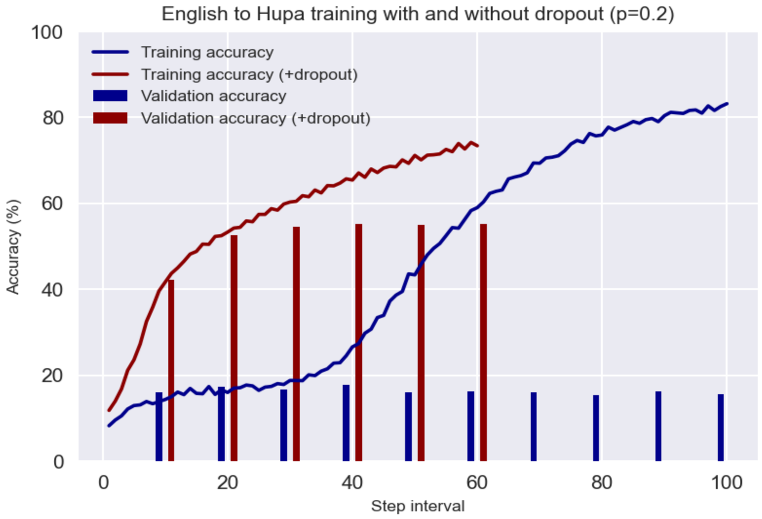
Hupa Results

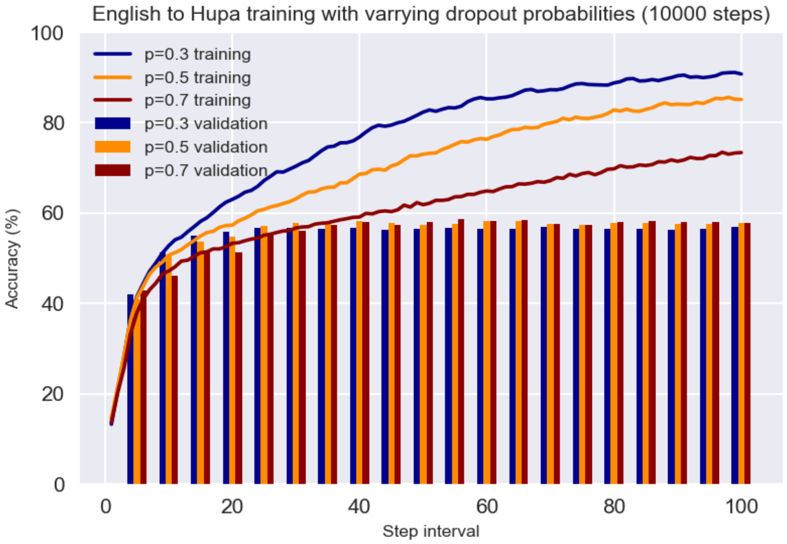
Training and Validation

Initially, before implementing sub-lexical tokenizing or dropout, the training and validation process for both translations directs were very similar, as seen in the graph below. Training accuracy improved greatly with time and reached up to ~80%. However, validation accuracy would become stagnant quite early and not improve, capping around ~15-~25%.



This trend of training accuracy increasing as validation accuracy stayed stagnant was still the case after implementing the SentencePiece tokenizer, which led us to hypothesize that we were overtraining the model with the training corpus. To alleviate this, we implemented a dropout probability to our configuration file before training. This parameter describes the probability that a node in the model will get dropped, which in turn avoids certain nodes from becoming too dependent on certain weights or previous nodes. This greatly improved the training process. While there was still a stagnation in validation accuracy, the upper limit was now closer to ~55% which the pre-dropout model never could achieve. Several different dropout probabilities were tested: 0.2, 0.3, 0.5, and 0.7. We found that increasing dropout probability only marginally increased validation accuracy movement and also decreased the upper limit of training accuracy, which is a logical result since the purpose of dropout is to avoid overtraining.





Interestingly enough, this similar training pattern seen with English-to-Hupa training was not the same when the same protocols were implemented in Hupa-to-English training. With a dropout probability of 0.7, 10,000 training steps, and the SentencePiece tokenizer, Hupa-to-English training only reached ~60% training accuracy and validation accuracy stopped growing at ~30%. The reason for this asymmetry between translation directions does not have a clear answer.

Translation Accuracy

Models were saved for both translation directions at different levels of the training protocol. English-to-Hupa models from 10,000-step training were saved at both 0.3 and 0.7 dropout probability. A Hupa-to-English model was saved also at 10,000 steps with 0.7 dropout probability. There were more models that were saved, but these come from training prior to adding dropout and are thus most likely overfitted to the training corpus.

Despite identical tokenizing and dropout protocols, English-to-Hupa translations were more accurate than Hupa-to-English. After 10,000 steps of training with 0.3 dropout probability, English-to-Hupa translations had an average predictive score of ~-0.22 and a BLEU score of 0.15. Increasing the dropout probability marginally improved both scores, decreasing the average predictive score to -0.43 and increasing BLEU score by 20% to 0.18. In the other direction, Hupa-to-English predictions were less accurate, which reflects its lower training and validation accuracy as well. 10,000 step, 0.7 dropout training yielded an average predictive score of -0.75 and a BLEU score of 0.05.

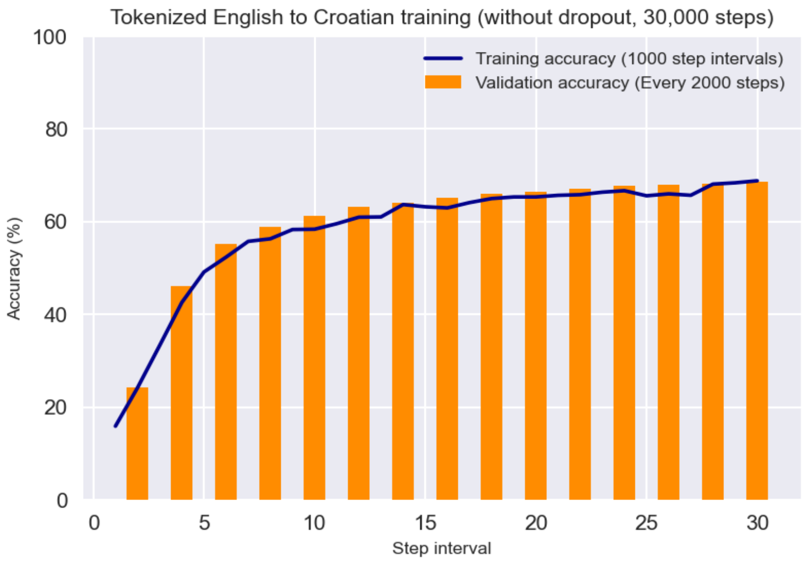
Translation predictions were largely inaccurate as well. Some predictions, particularly in the first two lines of the table below, do show possible evidence that not only is OpenNMT learning at the lexical level, but also at the morphemic level, with the accurate prediction of the 1st person singular possessive prefix *whi-*. It is important to note that the tokenizer separates this prefix in *whits'ine'* as *whits-*, which can ultimately limit accuracy if the productive morpheme is not isolated properly. However, most phrases were translated very inaccurately, without any evidence of lexical or morphemic learning, as exemplified by the last two examples.

English Test	**Hupa Prediction**	**Prediction Translation**
Yes, said Iris.	*Haya:ł 'a'de:ne'.*	Then she said it.
My hand	*Whits'ine'*	My bone/my leg
They both light them on the fire.	*Xontah-me' yehch'iwinde:tł'.*	They went into the house.
I ate acorns.	*Xona: ya'wehs'a'.*	His eye, he sat down.

Croatian Results

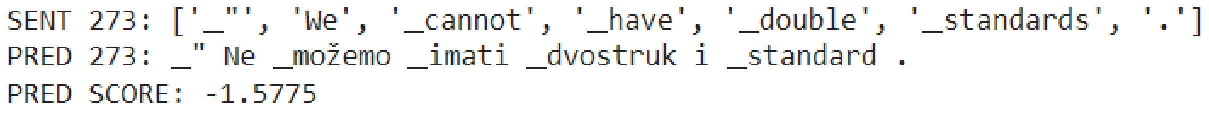
Accuracy

Unlike the model trained on Hupa-English data, the model for Croatian-English did not employ the dropout tool. The model was trained for 30,000 steps. At each step, accuracy increased while perplexity decreased with the end result of 68.82 for accuracy and 4.23 for perplexity. Similar numbers are reported for validation accuracy and validation perplexity. The accuracy sharply increased before tapering off. Had the model been trained over more steps or dropout probability been implemented, the accuracy may have continued increasing. Overall, the data collected provides figures better than or comparable to those reported by Klein et al. (2018) for German-English NMT.



Pred Score and BLEU

The prediction score for the translations do not yield much information about their quality. For English to Croatian, the pred score was between approximately -0.5 and -26. However, in manually going over some of the translations, they appear to be mostly accurate.



```
SENT 273: ['_"', 'We', '_cannot', '_have', '_double', '_standards', '.']
PRED 273: _" Ne _možemo _imati _dvostruk i _standard .
PRED SCORE: -1.5775
```

In the above figure, the translation is exact, with the exception of the plural marker being absent from the word "standards" in Croatian. Some common errors included small morphological mistakes like this one, in addition to leaving out subordinate clauses and swapping verbs that are semantically similar but not perfect synonyms of each other (e.g. "to welcome reforms" in English translated to "pozdraviti reforme" in Croatian which would be closer to "to greet reforms"). The pred score for Croatian-to-English translations was between approximately -3 and -50. Many words were OOV, resulting in incomplete translations. It is unclear why there were so many unknown tokens.

While calculating the BLEU score, a mismatch between the number of predictions and references was reported which may be the result of overfitting. In the future, a fine-tuned model may be trained on a different dataset in order to avoid the issue.

Impact

Every community of speakers, including those belonging to demographic groups historically underrepresented in the AI space, deserves language tools of the same caliber as other speech communities. While state-of-the-art language models have been launched for everyday use by speakers of widely-used WLs such as German and Arabic, less work has been done fine-tuning models for lower-resource languages. Tools such as Google Translate and Siri Translate are accessed by millions daily for translations ranging in size and importance from legal documents to restaurant orders. The translation needs of speakers of large WLs are the same as speakers of lesser-used languages. In addition, issues may arise as a result of poor translations. A company whose goal is to adapt their product to a new market would require high-quality language localization which may be aided by MT. In everyday conversation, a breakdown may occur if a speaker uses a poor translation in an attempt to engage with a speaker of another language. The authors of this paper experienced such a mishap recently when Google Translate spit out a Croatian greeting that has been obsolete since the fall of Socialism and is now considered taboo.

Future Directions

If work on this project were continued, more hidden layers could be added to the model which could result in higher accuracy. A higher dropout probability could be implemented, as well, since it showed promising results for the Hupa-English data. In a recent paper by Goldsmith and Mpiranya, an algorithm called *Linguistica* which has been developed by Goldsmith and others over the course of two decades was assessed on its learning of morphological structures in Swahili. The authors reported that while it does have trouble recognizing a possessive marker, it appears to have learned the structure of Swahili prefixes, correctly combining them with roots in a consistent manner. *Linguistica* could be implemented in OpenNMT for Hupa and Croatian as the code for it has been made available online.

References

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Goldsmith, J., & Mpiranya, F. (2022). Learning Swahili Morphology. *The University of Chicago*.

Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J., & Rush, A. M. (2018). OpenNMT: Neural Machine Translation Toolkit. arXiv preprint arXiv:1805.11462.

Klein, G., Zhang, D., Chouteau, C., Crego, J., & Senellart, J. (2020). Efficient and High-Quality Neural Machine Translation with OpenNMT. *Proceedings of the Fourth Workshop on Neural Generation and Translation*. https://doi.org/10.18653/v1/2020.ngt-1.25

Turovsky, B. (2016, April 28). Ten Years of Google Translate. Google. Retrieved from https://blog.google/products/translate/ten-years-of-google-translate/