CS 221: Project – Final Report
Machine Translation from English to Inuktitut: Data Augmentation

Christopher (Egalaaq) Liu (CS 221 & CS 229),
Brian Wang (CS 221 & CS 229),
Paul Magon de La Villehuchet (CS 221),
Yao Yang (CS 229)

Abstract

Our project attempts to translate English to Inuktitut using Neural Machine Translation (NMT) models, focusing on the impact of different levels of data augmentation on translation performance. Our results show for the task of machine translation, due to the constraint of having a limiting vocabulary size, data augmentation with unsupervised tokenization can hurt performance if too many out-of-vocabulary tokens are learned on auxiliary datasets.

Motivation

Automatic translation tools have been widely used in recent years to improve translation. Yet, automatic translation tools do not yet exist for endangered Inuit languages such as Inuktitut, due to limited training data available. Inuktitut is one of the 8 major Inuit languages of Canada. It is spoken by around 35,000 people in North Canada.

In addition to the challenge posed by the limited training data available, another major challenge in Inuktitut-English translation is the fundamental difference in the two languages. Inuktitut is a polysynthetic language; its words are made of multiple elementary units (morphemes), added as postbases to a root. Thus, a word in Inuktitut can be equivalent to a whole sentence in English. Example: *umiaqlikmut* = *umiaq* (boat) + *lik* (owner) + *mut* (to) → to the boat-owner

Our project will specifically look at how applying morpheme parsing and data augmentation to the Inuktitut input dataset improves English to Inuktitut translation. For CS 221, our experiment will specifically explore data augmentation. As a higher objective, since this is a novel application, we hope that an attention-based NMT model will provide a reliable translator.

Project Pipeline

Dataset

Hansard We mainly use the dataset published by the legislative body of Nunavut, Canada, with the help of Benoit Farley and the National Research Council of Canada (NRC). The dataset contains recordings from parliamentary proceedings from April 1, 1999 to November 8, 2007 and includes around 400,000 lines of parallel Inuktitut and English.

Bible For our data augmentation research question, we used a symbolic transcription of the Bible which was published in Inuktitut by the Canadian Bible Society in 2012.

Cleaning

Hansard To prepare the data for machine translation, we wrote scripts to clean and preprocess the data. The scripts accomplished tasks such as removing empty entries, non-ASCII characters, and unwanted entries caused by web-scraping when the dataset was originally created (i.e. web page artifacts). In addition, the dataset was paired down to a corpus with the 50,000 shortest Inuktitut entries in order to speed up the process of hyperparameter tuning.

Bible The Bible was also transformed and cleaned. We implemented a script that converts the symbolic text into Latinized text. The World English Bible was used for the corresponding English verse translations. The final step consisted of removing references and marking verses containing conversational speech for further pairing.

Data Tokenization

Data tokenization is an important part of pre-processing for machine translation. Indeed, neural networks can only learn a finite number of words and they will show poor performance if the size of the vocabulary is too large. Tokenization allows us to break down similar words into component morphemes and word stems.

BPE Parsing Byte-pair encoding (BPE) is an unsupervised tokenization method to learn a vocabulary from a dataset. In BPE, the vocabulary is first initialized with all individual characters and then extended repeatedly by merging the most frequent pair of existing vocabulary tokens found in the dataset (7).

BPE can be applied to source and target datasets individually or jointly. Jointly-applied BPE learns the encoding of the union of the vocabulary of both languages. This

Department of Computer Science,
Stanford University, 2017

improves consistency between the source and target segmentation, in case jointly used tokens are segmented differently between both languages.

We applied joint BPE encoding to our datasets. We also experimented with various vocabulary sizes in order to understand the level of tokenization that produces the highest accuracy.

Machine Translation

Recurrent Neural Networks Historically, Statistical Machine Translation (SMT) has been the most widely-used and studied machine translation method. However, Recurrent Neural Networks (RNN) have recently been shown to match performance of SMT systems. RNN are a series of Neural Networks being fed by their predecessor and feeding their successor.

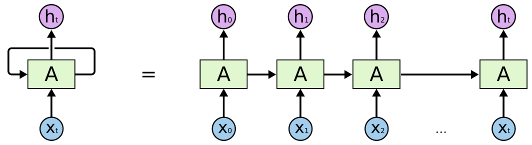

Figure 1: Basic Structure of a RNN

RNN maintains an internal state, such that the information from previous states can be used to inform future performance.

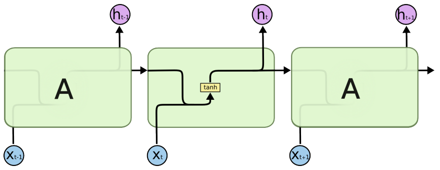

Figure 2: Internal Structure of a RNN

In our case, we used Bidirectional Neural Nets. BRNN differ from RNN by introducing a second hidden layer that flows the other way. However, RNN have one major shortcoming: RNN cannot learn information over long distances. Only recent information can be learned.

LSTM Long Short-Term Memory RNN are a particular type of RNN. They have the same structure than RNN with one major difference: the existence of a cell state, keeping information along the way. The information carried by the cell state is updated at each neural network.

Attention Mechanism The attention mechanism is a way to focus on different parts of the input sentence based on what the model has produced so far and is effective in sequential encoding and decoding tasks such as machine translation (3). This mechanism allows us to address the challenges that arise from Inuktitut-English word alignment. It does not rely on hard-coded rules and is completely automated.

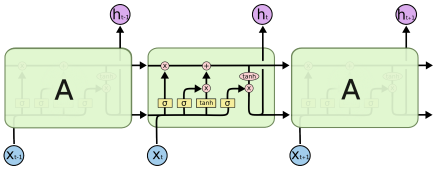

Figure 3: Internal Structure of a LSTM

Seq2seq We used a RNN with an attention mechanism using the *seq2seq* encoder-decoder framework from *Tensorflow* (4).

Evaluation Metric: BLEU Score

Bi-lingual evaluation understudy (BLEU) score is a metric for a predicted translation with a reference translation. The simple unigram BLEU score is computed by taking the maximum total count of each predicted word in any of the reference translations, summing the counts over all all unique words in the predicted translation, and dividing the sum by the total number of words in the predicted translation. For evaluation of our experiments, we use a modified BLEU score analysis as implemented in Moses. The multi-BLEU implementation in Moses takes the weighted geometric average of n-gram BLEU-scores $(n = 1, 2, 3, 4)$, and multiplies that result by an exponential brevity penalty factor.

$$BLEU = BP \cdot \left(\sum_{n=1}^{N} \frac{\log(P_n)}{N} \right)$$

Here P_n denotes the precision of n-grams in the hypothesis translation, and the N we use here is 4.

$$P_n = \frac{\sum_{n-gram \in hyp} \text{count}_{clip}(n - gram)}{\sum_{n-gram \in hyp} \text{count}(n - gram)}$$

$$BP = \begin{cases} \exp\left(1 - \frac{|\text{ref}|}{|\text{hyp}|}\right) & \text{if } |\text{ref}| > |\text{hyp}| \\ 1 & \text{otherwise} \end{cases}$$

Experiments

We completed three separate English-Inuktitut dataset comparisons.

(1) Hansard (shared BPE)
(2) Hansard + Bible(conversational only) (shared BPE)
(3) Hansard + Bible(full) (shared BPE)

The first applied shared BPE parsing on both source and target languages for the Hansard dataset, as recommended by Google's NMT System (4). The second included both the Hansard and conversational verses of the Bible. The third included both the Hansard and the full Bible. Shared BPE parsing was also completed on both

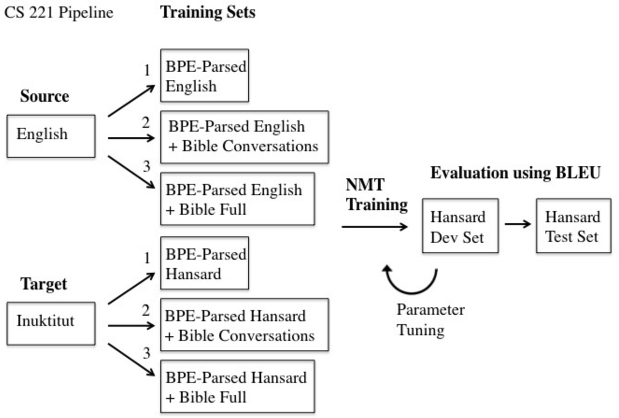

Figure 4: Pipeline of CS221 Experiments

source and target languages for the Bible-appended datasets.

These three experiments were specifically chosen to determine performance after including different types of training data. The second experiment seeks to answer whether including a domain-sharing (conversational) dataset will improve the translator. The third will explore performance when including more data, however, with reservation due to the inclusion of narrative style text on a primarily conversational training set.

Results and Analysis

Parameter Tuning

Results To explore parameter tuning, we chose to use the minimum parsing dataset containing the 50,000 shortest lines of Inuktitut along with he corresponding English translations for these lines. This dataset was randomly split into training, dev (validation), and test sets, with 45,000/2,500/2,500 lines respectively. The source-target direction was Inuktitut-English.

Based on TA recommendations, we conducted experiments varying dropout input keep probability, decoder number of layers, and decoder number of neural units. The model parameters we used for our baseline and parameter tuning experiments as well as experiments are detailed below in Table 1.

	Dropout	Decoder	Decoder		
Source	Input Keep	# of	# of	BLEU	@Step
Reverse	Probability	layers	units	Score	
False	0.8	1	128	11.84	50,000*
True	0.6	1	128	12.84	149,000
True	0.4	1	128	11.16	128,000
True	0.8	2	128	11.97	66,000
True	0.6	2	128	12.51	72,000
True	0.6	2	256	11.75	53,000

*: default

Table 1 : Summary of all the experiments for parameter tuning

Analysis Based on the highest accuracies and the high variability of BLEU scores, no tuning combination appeared to significantly outperform the others. Because of this, we chose to proceed with the default set of parameters for our full experiments, with one change to include source reversing, a decision we made based on previous work that has shown source reversing to improve machine translation performance.

It was difficult to assess differences between the parameters because BLEU scores tended to fluctuate so that the results had significant overlap in range. We hypothesize this might have been due to the size of the dev and test sets, and that increasing the dev and test set size would have helped to stabilize the BLEU score evaluations. In fact, for our full results, we used dev/test sizes of 5,000 and did not see as much fluctuating in BLEU score evaluations of predicted outputs.

Initially, we also explored tuning of the learning rate parameter, but we found that this was not necessary due to our use of the Adam optimizer. Adam adaptively updates learning rates for individual parameters by using moving averages of the mean of variance of past gradients, incorporating an idea similar to stochastic gradient descent with momentum. This optimization strategy enables the algorithm to converge quickly, even if starting from a larger learning rate (8).

Data Augmentation

Results We completed the base experiment and the two augmented experiments with the parameters described in the preceding section. The following plot (Fig. 5) shows that the Hansard-only (base) dataset and convo-augmented dataset exhibited a similar BLEU prediction score, while the Bible-augmented dataset saw a drop in BLEU prediction performance. The final test BLEU scores for the base/convo-augmented/Bible-augmented experiments were 2.55/2.75/0.34, respectively (Table 2).

Dataset	Dev BLEU	@Step	Test BLEU
Hansard only	2.22	59,000	2.55
Conversational	2.34	59,000	2.75
Both (BPE)	0.23	60,000	0.34

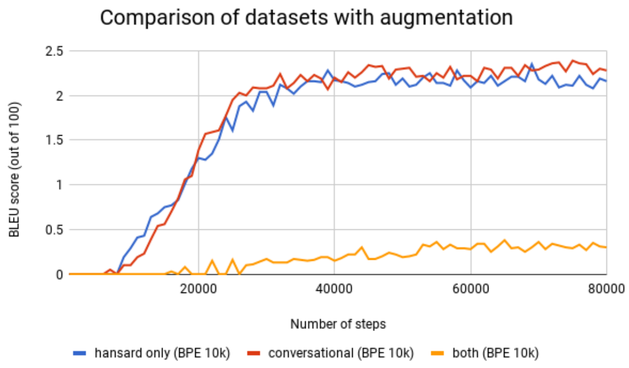

Figure 5: Comparing results after Hansard augmentation

Table 2 : Dev and test BLEU scores for the base and data-augmented experiments

Analysis While we were originally expecting the data-augmented dataset to exhibit similar or improved BLEU prediction scores, we hypothesize that the Bible-augmented experiment exhibited poor performance due to the presence of common tokens in the Bible dataset that do not occur in the Hansard dataset.

Since we were constraining our vocabulary size to 10,000 merges only, learning more common Bible-specific tokens at the expense of less-common Hansard tokens would hurt performance. We suspect that the convo-augmentation did not cause BPE to learn many convo-specific tokens, while the Bible-augmentation did, which explains the differences in performance. The following sentences are examples of experiment output:

original
uqaqti tusaajikkut piqujivungaarutitaqappuq
hansard
uqaqti tusaajikkut piqujivungaaruti pigiaqtitauvuq
convo
uqaqti tusaajikkut pigiaqtitaujuq naammattuq
both
uqaqti tusaajikkut aippanga ininga

Further analysis of the combined dataset

Results Based off of our results in the preceding section, we decided to explore the effects of increasing the BPE vocabulary size, to understand if an increased number of merges would prevent the adverse effect of Bible-augmentation on BLEU scores, by allowing BPE to learn both common Bible-specific tokens and less-common Hansard tokens under the larger vocab size. In addition, we experimented with an unparsed dataset as well.

The final test BLEU scores for unparsed/10k/30k vocabulary (BPE) sizes were 0.64/0.34/2.81 respectively (Table 3).

Dataset	Dev BLEU	@Step	Test BLEU
10k Both	0.23	60,000	0.34
32k Both	2.40	81,000	2.81
Unparsed Both	0.52	48,000	0.60

Table 3 : Comparison of scores for the further analysis of combined (both) dataset

Analysis Our results are consistent with our hypothesis that a larger BPE vocabulary size prevents the adverse effect of Bible-augmentation on BLEU scores that is exhibited when using a BPE vocabulary size of 10k.

Supplementary Results and Analysis

Attention Mechanism Proof-of-Concept

Results The following figure demonstrates that using an attention mechanism improved our model. For this comparison, default parameters were used with English as the source and Inuktitut as the target language.

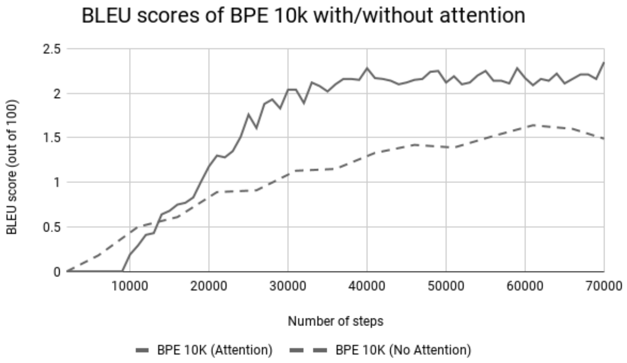

Figure 6: BPE results with and without attention

Analysis Using an RNN with an attention mechanism extends the model by allowing it to automatically (soft)search for parts of a source sentence that are relevant to predicting a target word. Our results are consistent with previous research suggesting that using an attention mechanism will increase translation performance (1).

Beam-Search

Results One way to improve prediction is through usage of beam search, which considers 5 ”beams” of candidate predictions at each step instead of a greedy approach. Our test BLEU scores on BPE-10k with and without beam search were 10.17/10.05, respectively (see Table 4).

Dataset	Dev BLEU	@Step	Test BLEU
Without Beam	10.00	60,000	10.05
With Beam	10.01	60,000	10.17

Table 4 : Dev and test BLEU scores on BPE-10k with and without beam search

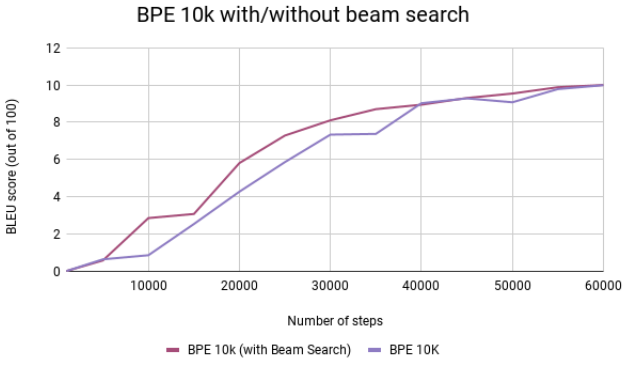

Figure 7: BPE results with and without Beam-Search

Analysis In our experiment, Beam-Search did not make significantly better predictions; however, it took a significantly longer time to generate a prediction due to the additional searching time. Without additional computational resources, we would not choose to use beam search due to the additional increase is computational cost.

Batch Number

Results An increased batch size allows for better parameter updates due to more accurate gradient approximation, as the sum of the batch gradients are averaged over the size of the batch. With an increasingly large number of parameters to update in deep neural networks, we hypothesize that batch parameter updates may be increasingly important to help the parameters update in the right direction. We ran two experiments of the max-parsed data using default parameters with different batch sizes of 32 and 128 to explore the differences in performance of varying batch size.

This plot was generated using data from our CS 229 project, but contains results that are useful for this study. Please refer to our CS 229 paper for further details about data used.

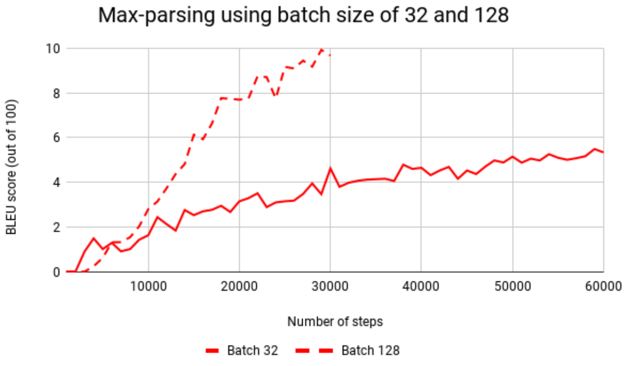

Figure 8: Batch size comparison

Dataset	Dev BLEU	@Step	Test BLEU
Batch 32	5.49	59,000	5.90
Batch 128	9.93	29,000	10.16

Table 5 : Dev and test BLEU scores for max-parsed data using batch sizes 32 and 128 (CS 229)

Analysis Upon inspection of our plot, step 40,000 of the batch-32 experiment is only slightly better than step 10,000 of the batch-128 experiment, suggesting that at least for our experiments, there is no evidence that batch-128 updates are significantly better than batch-32 updates or vice versa.

Conclusion and Future Work

With the intention of further research and delivering a web-based translation tool, we specifically chose this project in order to find the best strategies for improving Inuktitut translation. Our research question for CS 221 examines how data augmentation affects machine translation from English to Inuktitut.

In summary, we conclude that the use of byte-pair encoding can pose challenges when applied to smaller augmented sentences containing mixed-domain text (e.g. conversational and narrative.)

As a potential future experiment for this study, we might experiment with only learning BPE on the base (Hansard-only) dataset, applying the learned vocabulary on the augmented datasets and seeing if this improves BLEU predictions scores. Another future experiment is trying to apply a rule-based parsing strategy instead of BPE on the augmented datasets.

As additional future work, we hope to apply the lessons learned from both CS221 and CS229 experiments on similar agglutinative Eskimo/Inuit languages. Our next goal is to apply this to Central Yup'ik Eskimo, a language primarily spoken in Southwest Alaska!

Complementary CS 229 Project

In addition to the CS 221 project, we conducted a complementary project in CS 229 that compares translation quality across various parsing strategies, including rule-based and unsupervised learning-based parsers.

Rule-Based vs. BPE

The following Inuktitut to English translation experiments were completed using default parameters as described in our CS 229 paper:

1) Minimum parsed Inuktitut & Unmodified English
2) Maximum parsed Inuktitut & Unmodified English
3) BPE-parsed Inuktitut & BPE-parsed English (joint)

The following plot shows that BPE-parsed data with 10k merges outperformed both rule-based parsing datasets in translation accuracy.

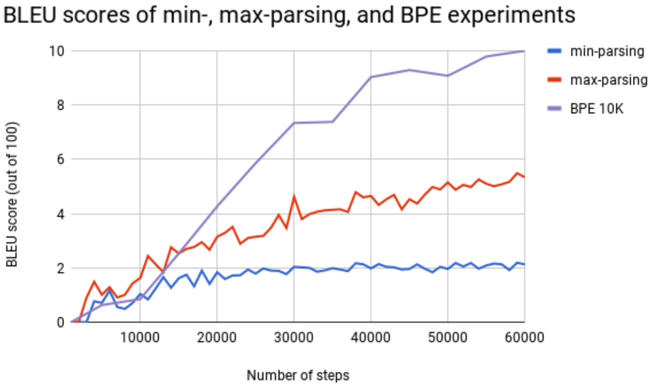

Figure 9: Min/max rule-based and BPE 10k parser results

Application Limitations Although the BPE at 10,000 merges performed best in our CS 229 project, this experiment reveals a trade-off when using BPE with datasets containing mixed domains. If all the data comes from a single source, using BPE may be a reasonable first strategy to achieve higher translation accuracy.

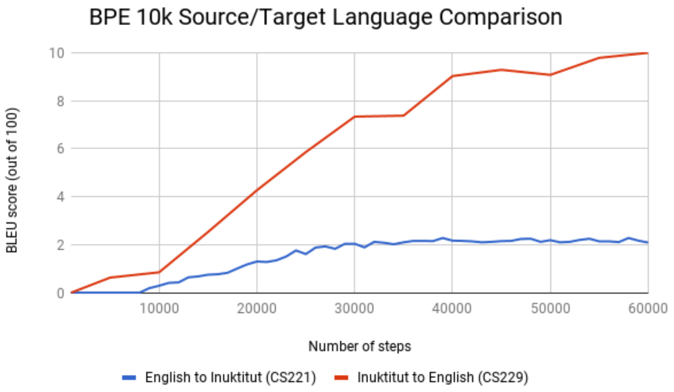

Figure 10: BLEU score v. number of steps comparing language direction

Comparing source/target direction Our results were notably different after reversing language direction for BPE with 10k. For both datasets, word segmentation occurred more in the Inuktitut dataset than in the English dataset. Thus, we hypothesize that the difference in results is due to the increased difficulty of returning an accurate prediction for a more segmented language. For example, it might understandably be more difficult to predict 30 tokens and place them in the right order than it would to be predict 10 tokens.

The task of translating from English to Inuktitut requires not only the prediction of more total tokens but also the prediction of more tokens in the correct order, to be rejoined after post-processing. In addition, output order of predictions is also critical to correct Inuktitut translation, as increased segmentation also increases the possibility of incorrect rejoining of segmented predictions, which would not occur in a less segmented language, which is more likely to output full tokens that do not need to be rejoined. We hypothesize that the significantly increased amount of segmentation in Inuktitut is the main factor contributing to the difference in performance for prediction between the two directions.

References

(1) Bahdanau, D.; Cho, K.; Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Proc. International Conference on Learning Representations

(2) Cho, K.; Van Merrinboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation

(3) Briggs, J.; Johns, A.; Cook, C. (2015) Utkuhiksalingmiut Uqauhiitigut: Dictionary of Utkuhiksalingmiut Inuktitut Postbase Suffixes. The postbase dictionary we can use to build our own limited parser

(4) Kalchbrenner, N.,Blunsom, P. (2013). Recurrent Continuous Translation Models. In EMNLP (Vol. 3, No. 39, p. 413)

(5) Kingma, D., Ba, J. (2014). Adam: A method for stochastic optimization

(6) 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. (pp. 311318)

(7) Sennrich, R.; Haddow, B.; Birch A. (2016). Neural Machine Translation of Rare Words with Subword Units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (pp 17151725)

(8) Seq2Seq Framework
https://google.github.io/seq2seq/

(9) Sutskever, I.; Vinyals, O.; Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112)

(10) The UQAILAUT Project. Accessed Oct 20, 2017. http://www.inuktitutcomputing.ca/Uqailaut/info.php

(11) Yonghui W. et al. (2016) Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation