Neural Morphological Analyzer for Inuktitut

Johanna Männistö

3 January 2023

1 Introduction

In this report, I detail the steps taken to train a neural morphological analyzer for Inuktitut. This paper will cover the data and preprocessing steps taken to prepare the data for training an RNN model, the steps taken and parameter changes made to improve the model, in addition to the various metrics used to evaluate the model. I found that the best performing model was a monolingual bidirectional LSTM model trained for 10,000 steps. This model achieved an overall accuracy of 76% with limited errors when they occur.

2 Background & Motivation

2.1 About Inuktitut

Inuktitut is a polysynthetic language of the Eskimo-Aleut language family. It is spoken in Eastern Canada and is an official language of the territory of Nunavut, Canada. As of 2016, there were approximately 40,000 speakers of Inuktitut.

Most words in Inuktitut are composed of multiple morphemes. Each word begins with a root, either a verb, noun, or demonstrative root or an interjective. These roots may be followed by affixes and then grammatical endings which can express a variety of grammatical or semantic meanings. These morphemes also exhibit fusional properties, denoting multiple grammatical or semantic meanings within one morpheme.

Polysynthetic languages like Inuktitut can express with one word what other, more analytic, languages require a phrase for. This can be seen in the example originally provided by Micher 2018 [5] *Qanniqlaunngikkalauqtuqlu aninngittunga*, translated into English as "And even though it's not snowing a great deal, I am not going out". The full gloss for this sentence is provided below where one can see the multiple morpheme construction.

Qanniqlaunngikkalauqtuqlu	aninngittunga
qanniq-lak-uq-nngit-galauq-tuq-lu	ani-nngit-junga
snow-a_little-frequently-NOT-although-3.IND.S-and	go_out-NOT-1.IND.S
"And even though it's not snowing a great deal"	"I'm not going out"

2.2 Why an RNN?

Frequently, Finite State Technology (FST) are used to create morphological analyzers and generators for languages. They have the benefit of mostly requiring the linguistic knowledge instead of a large amount of data required by neural models. They also can switch between being an analyzer and generator with the same system, while two separate systems must be trained when using neural architectures. However, FST systems rely on a lexicon to provide analyses for inflected forms and are unable to provide analyses when lemmas or morphemes are not accounted for in either the lexicon or rules. If an FST model comes across unknown morphemes, roots or a combination thereof it will fail to return an analysis.

FST models are also time consuming to build. They require significant knowledge of the language as well as of the tools for finite state modeling. Moeller et al., 2018 [7] described this as a Pareto-style trade off where while much of the grammar could be developed quickly, there is a long tail end of effort needed to keep up with lexicon expansion and to manage difficult cases. Even endangered languages expand, change and shift - they need a morphological analyzer that can generalize to unseen forms to account for imperfect construction and for language evolution.

Neural Morphological Analyzers require data, which is difficult to acquire for many of the world's languages. However, there are some resources available for Inuktitut, namely the Nunavut Hansard Corpus and the Uqailaut Analyzer. The Legislative Assembly of Nunavut publishes a parallel corpus of its proceedings in English and Inuktitut. The first version of this corpus was published in 2003, and the most recent version, version 3.0, was published in 2020 making approximately 1.3 million aligned English-Inuktitut sentences available. In 2009, a finite state transducer, morphological analyzer, called the Uqailaut Analyzer was developed by the Institute for Information Technology in Canada. This tool can provide the morphological decomposition of an inputted Inuktitut word.

Given the availability of resources and data for Inuktitut, and the lack of generalizability in FST models, training a neural model to perform morphological analysis on Inuktitut could achieve better generalizability and improve performance in unseen situations which is currently not feasible with the Uqailaut Analyzer and other FST systems.

3 Data

The data used for this project came from the data set created by Jeffrey Micher's work of morphologically analyzing the Inuktitut words in the Nunavut Hansard corpus [4]. This served as supervised examples to train a recurrent neural network (RNN) encoder-decoder. The morphological analysis of the corpus words were produced by the Uqailaut Project analyzer. The data in these files was in the following format:

<SURFACE FORM WORD> <TAB> {morph1}{morph2}...{morphx}

with a pipe "|" as a word analysis boundary. Each morpheme analysis contained the following structure:

{SURFACE_FORM:LOWER_FORM/TAG}

This project used the uqailaut.2.first.analysis.tar.gz data set prepared by Micher 2018 [6] which contains a total of 750,493 unique words. Each word, however, is only provided one analysis, despite that Inuktitut words frequently have have at least two salient analyses [6]. Given the project scope, I focused on generating only one output.

3.1 Morpheme Tags

Table 1 shows a summary of the morpheme tags that appear in the data set.

Inuktitut Analyzer <type>Tags

Tag	Description	Tag	Description
v	verb root	a	adverb
n	noun root	c	conjunction
q	tail suffix	nn	noun-to-noun suffix
nv	noun-to-verb suffix	vv	verb-to-verb suffix
rad	demonstrative adverb root	rpd	demonstrative pronoun root
tv-<mode>-<subject prs & num>[-<obj prs & num>-[prespas|fut]			verb ending
tad-<case>-<number>[-<possessor prs & num>]			noun ending
tpd-<case>-<number>			demonstrative pronoun ending
pd-<location>-<number>			demonstrative pronoun
ad-<location>			demonstrative adverb

Inuktitut Analyzer <mode>tags

Tag	Description	Tag	Description
dec	declarative	ger	gerundive
int	interrogative	imp	imperative
caus	causative	cond	conditional
freq	frequentative	dub	dubitative
part	participal		

Inuktitut Analyzer <case>Tags

Tag	Description	Tag	Description
nom	nominative	acc	accusative
gen	genitive	dat	dative
abl	ablative	loc	locative
sim	similaris	via	vialis

Inuktitut Analyzer <number>Tags

Tag	Description	Tag	Description
<nb>	integer, number or person	s	singular
d	dual	p	plural

Inuktitut Analyzer <location>Tags

Tag	Description	Tag	Description
sc	static or short	ml	moving or long

Table 1: Inuktitut Morpheme Tags, Taken From Micher 2018 [6]

3.2 Preprocessing

Below is the list of steps taken to clean and process the data, I used the script IO.sh (included in the associated attachments) to perform the listed actions. Each surface form word had a tab deliminator separating it from the morphological analysis and there were no punctuation or missing analyses either from the set.

- shuffle the data
- split the data into a surface form file and morpheme tag file
- tokenize the file of Inuktitut words

After performing these steps, I continued the processing with the script dataClean.py which takes in the morpheme file from the steps above. This script is also included in the attachments and it cleans up the data a bit more with the following steps:

- remove the pipe "|" denoting the end of the morphological analysis
- remove the surface form in the morphological analysis
- remove deep forms of all morphemes but the root
- tokenize lemmas and tags by character, where morpheme tags are single characters
- split the data into train-validate-test sets

After cleaning the data, I finally split the data into the train-validate-test sets using the script datasplit.sh, included in the attachments.

After preprocessing, I used OpenNMT-py's [3] *onmt_ build_ vocab* tool to build the vocab files for both the source and target where *n_ sample -1* to capture all the vocabulary.

4 Neural Morphological Analyzers

Like many others, I approached process of morphological analysis as machine translation task: given an input or source (i.e. the surface form word) translate to the target (in our case, the morphological analysis). Specifically I used an encoder-decoder architecture to encode the surface form of the word and decode it into the deep form of the root with morphological tags.

4.1 Monolingual Models

I trained and evaluated three monolingual models for this project. Each model was implemented using OpenNMT-py [3] version 2.3.0 with 1 layer, an RNN and word_vec size of 256 and a batch size of 64. The models were trained for 10,000 steps. Each model used the same training set of size 50,000 and the same validation and test sets, each with a size of 5,000. Specific parameters by model are detailed in Table 2.

Model 1	**Model 2**	**Model 3**
Model Type: LSTM	Model Type: LSTM	Model Type: Bidirectional LSTM
Attention: None	Attention: General	Attention: General

Table 2: Model Specific Parameters

Following high performance from other related tasks for morphological inflection (Kann et al., 2017 [2]) and for neural morphological analyzers for another polysynthetic language (Moeller et al., 2018 [7]) my best performing model was a bidirectional LSTM sequence-to-sequence (seq2seq) model with an attention mechanism.

4.1.1 Train-Validate-Test

The data was split into 3 different data sets: train, validate, and test. Table 3 shows the line counts for each file type.

File Type	Lines
train	50,000
val	5000
test	5000

Table 3: line and character count of train, test, and validate files

4.2 Multilingual Models

Frequently adding multiple languages, or in this case morphological analyses of multiple languages, into one system can improve performance. Multilingual models benefit from transfer learning and reduce the risk of overfitting with the presence of more and different data.

4.2.1 Data

As data for other Inuit languages' morphological analyses are unavailable, I decided to use an agglutinative with polysynthetic characteristics language: Navajo. The UniMorph project has many datasets of annotated morphological data in a universal schema that has been used for several SIGMORPHON shared tasks over the past few years. The UniMorph Project had a data set for Navajo [1] which included different noun and verb paradigms, however it did not include any adjective paradigms.

The data was downloaded in a UTF-8 encoded format, where each line included a lemma, surface form, and morphological tags. The morphological tags are in the UniMorph format, separated by semicolons. Each column was separated by a tab. An example from Navajo is included below:

akágí yikágí N;PSS4;ARGAC3S

This data set had a total of 11883 data points, much smaller than the available data for Inuktitut

4.2.2 Processing

Given that the data set for Navajo followed the UniMorph schema while the Inuktitut data set followed another paradigm, I transformed the Inuktitut data set to correspond with the UniMorph schema. Following the specification of the schema as described in Sylak-Glassman 2016 [8], I converted the Inuktitut morphs tags to similar tags in the UniMorph schema using the unimorphize.py script (see attachments). Table 4 shows the correspondence between the original tags (as detailed in Table 1) and the newly assigned UniMorph tags. To prepare the data for the multilingual model the following steps were taken using the multilingualprocess.sh script (see attachments):

- converted Inuktitut data set to UniMorph schema
- rearranged Navajo data set columns to surface form, lemma, tags to match Inuktitut ordering
- added language tags in front of the corresponding words
- combined languages to create vocabulary
- shuffled data
- tokenized each file by character, where morpheme tags are single characters using charToken.py (see attachments)

- split the data set into words and morphemes
- split data into train-validate-test sets

Corresponding Inuktitut and UniMorph tags

Inuktitut Tag	Unimorph Tag	Inuktitut	UniMorph Tag
v	V	a	ADV
n	N	c	CONJ
q	Q*	nn	NN*
nv	NV*	vv	VV*
rad	ADV	rpd	PRO
tv-<mode>-<subject prs & num>[-<obj prs & num>-[prespas|fut]			<mode>;<subject prs & num>[;<obj prs & num>; PRESPAS | FUT
tad-<case>-<number>[-<possessor prs & num>]			<case>;<number>[;<possessor prs & num>]
tpd-<case>-<number>			PRO;<case>;<number>
pd-<location>-<number>			PRO;<location>;<number>
ad-<location>			ADV;<location>
dec	DECL	ger	PROG
int	INT	imp	IMP
caus	CAUS	cond	COND
freq	FREQ	dub	DUB*
part	PART	via	BYWAY
nom	NOM	acc	ACC
gen	GEN	dat	DAT
abl	ABL	loc	LOC*
sim	EQTV	<nb>	#.
d	DU	s	SG
p	PL	sc	SC*
ml	ML*		

Table 4: Corresponding Inuktitut Morph Tags and UniMorph Tags

4.2.3 Train-Validate-Test

As with the monolingual models, the combined Inuktitut-Navajo dataset was split into train, validate, and test sets. Below is the breakdown of the lines counts for each file type.

File Type	Lines
train	51,884
test	5000
val	5000

Table 5: line and character count of train, test, and validate files

In the training data, 9990 samples were of Navajo and the other 41894 were Inuktitut. In the validation set, 4024 samples were Inuktitut and 976 were Navajo. The test set was performed entirely on Inuktitut.

5 Neural Morphological Generators

I also decided to try creating a generator for Inuktitut given a root and series of morphemes. I used the same set up as Model 3. The model was implemented using OpenNMT-py [3] with 1 layer, a bidirectional LSTM with general attention an RNN and word_vec size of 256 and batch size of 64.

This model used the same data as the monolingual models, except using the morpheme data set as the source and the word set as the target.

6 Results

Models were analyzed using three metrics:

1. **Overall Accuracy:** This compares the entire predicted stem plus morpheme tags to the expected output. Any errors in the stem or tags would mark it as incorrect
2. **Average Levenshtein Distance for Lemma:** This compares the similarities between the predicted and actual lemmas. Smaller values indicate higher accuracy and higher values indicate poorer performance.
3. **Average Levenshtein Distance for Morpheme Tags:** Similar to above, this compares the similarities between the the predicted morpheme tags and actual morpheme tags. Each incorrect morpheme tag is considered to be a distance of 1 away from the morpheme tag prediction. Lower values indicate better model performance.

The results of these metrics are summarized in Figure 1 and Figure 2. Scripts for the model's performance evaluation is included in Appendix # and attachments under metrics.py.

Figure 1 shows the overall accuracy of each model. Notably we can see in this figure that the multilingual model analyzes none of the test words perfectly while the other models achieved between 61% and 76% accuracy.

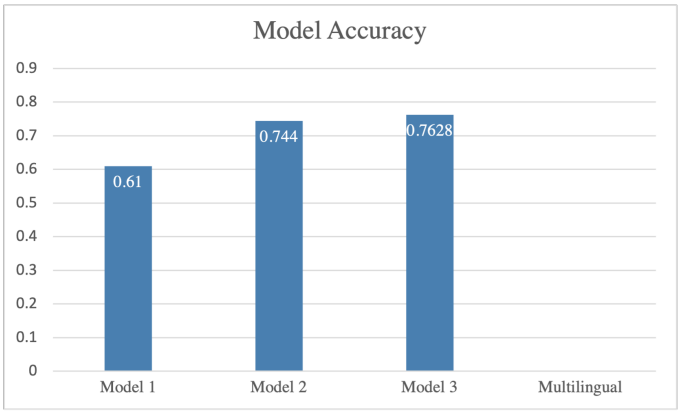

Figure 1: Total Accuracy by Model

Figure 2 shows the average Levenshtein distance between the model prediction and actual for the predicted lemma and predicted morphs. Although the multilingual model received the worst score in total accuracy, it still outperforms Model 1 with both average Levenshtein distances indicating that when Model 1 does have incorrect predictions the multilingual model's predictions are closer to the expected results than the Model 1's prediction (on average).

We see in both figures that Model 3, the model using a bidirectional RNN architecture with general attention achieves the best results. This model makes on average 0.3254 errors on a predicted stem, and 0.6176 errors on the series of predicted morphs, but predicts the correct analysis 76% of the time.

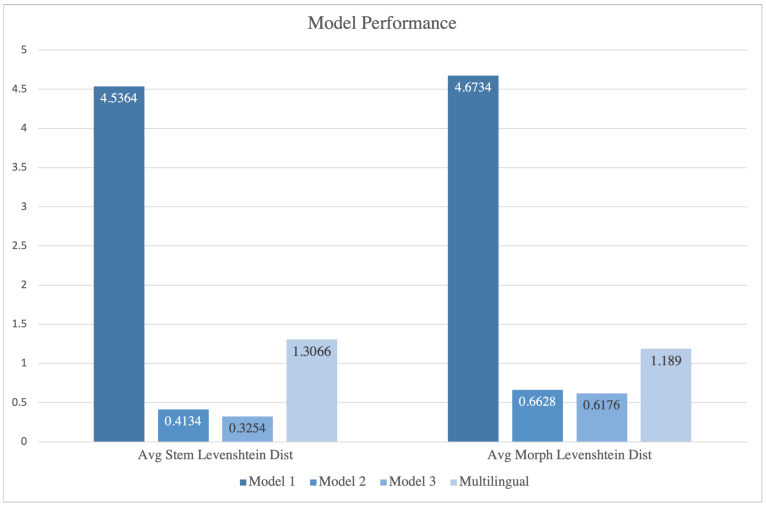

Figure 2: Average Levenshtein Distances by Model

The generator model also had an overall accuracy of 0%. The average Levenshtein Distance between the predicted, or generated, word and the expected word was 14.09. This very high as it was nearly the length of the average word in the test set, 17.02. Word lengths in the set ranged from 3 and 40 characters long. This poor performance is likely due to a scarcity of information provided to the model. The data used only included the word root, typically either a noun or a verb, but there are many other lexical morphemes included in most words which may share morpheme tags but differ semantically. Including all semantic morphemes in the source data for the generator should improve the generator's ability to make accurate predictions. It could be that the generator is able to accurately predict a grammatical word given a root and morpheme tags, it just is unable to predict the specific word with additional semantic meaning given the complex and fusional nature of Inuktitut.

7 Discussion

The improvements on the monolingual model were as predicted, once attention was added there was an immediate improvement to the models' ability to predict the correct analysis - dropping from Levenshtein distances of 4.5 and 4.6 between the predicted and actual lemmas and morphs respectively to only 0.4 and 0.6. Shifting to a bidirectional LSTM yielded some additional performance improvements as well, but not as significantly as adding attention.

Incorporating multiple tasks (or morphological analyses for multiple languages) into one model can yield improvements, especially on the lower-resourced languages. Training a multilingual morphological analyzer for both Inuktitut and Navajo did not appear to improve performance on the Inuktitut data, rather it deteriorated the performance of the analyzer compared to the monolingual models.

It is possible that these languages are too distant and aren't able to benefit one another through transfer learning as much as hoped. Both languages belong to different language families and despite their polysynthetic characteristics, are quite different from one another. There was also a large

difference in size between each data set. A more balanced data set could possibly improve results. Incorporating weights into the data, to ensure that the Navajo data set isn't only 20% of the size of the Inuktitut data, could potentially improve performance. However, this imbalance doesn't fully explain the overall drop in model performance as most of the data within the combined data set was Inuktitut. It is possible that the imbalanced data set along with the reduction in amount of Inuktitut data negatively impacted the model's performance, although it is likely not the sole reason. The multilingual model could also require additional training steps. As the model approached 10,000 steps the perplexity and accuracy reported each step was still changing and hadn't quite plateaued yet.

Despite the multilingual model's inability to perfectly analyze any of the test data set, we do see some improvement in the Levenshtein distances for both the lemmas and morphs. This indicates that although the model is unable to completely accurately predict the lemmas, it gets close. For Inuktitut, the words and number of morphs per word is extremely long, there are significant opportunities to make an error given the sheer length of words and number of morphemes. The average word length in the training data set,is 19.94 characters long, with a range from 2 to 53 characters. The average lemma length was 5.31 characters, ranging from 2 to 14 characters long and number of morphs to predict averaging at 5.53 morphs, ranging from 1 to 18 morphemes. For the test set, the words were between 3 and 40 characters, with an average length of 17.02. The lemmas range from 2 to 14, averaging 5.83 characters long. The number of morphemes tags per word is between 1 and 15, averaging 4.5 morphemes per word.

This project was an interesting endeavor and highlighted the importance of good data. Inuktitut is fairly well resourced due to tools already created and maintained but many other languages lack these resources. FST systems, while they have have drawbacks, also provide opportunities for linguists or other language experts to contribute to tool and data creation for lower resourced languages, allowing them to thus be better served in the age of data-driven technology.

References

[1] UniMorph Navajo Data.

[2] Katharina Kann, Ryan Cotterell, and Hinrich Schütze. Neural Morphological Analysis: Encoding-Decoding Canonical Segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 961–967, Austin, Texas, 2016. Association for Computational Linguistics.

[3] 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.

[4] Jeffrey Micher. Data Set of Morphologically Analyzed Inuktitut Words.

[5] Jeffrey Micher. Addressing Challenges of Machine Translation of Inuit Languages. Technical Report ARL-TN-0924, US Army Research Laboratory, October 2018.

[6] Jeffrey Micher. Provenance and Processing of an Inuktitut-English Parallel Corpus Part 1: Inuktitut Data Preparation and Factored Data Format. Technical Note ARL-TN-0923, US Army Research Laboratory, October 2018.

[7] Sarah Moeller, Ghazaleh Kazeminejad, Andrew Cowell, and Mans Hulden. A Neural Morphological Analyzer for Arapaho Verbs Learned from a Finite State Transducer. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 12–20, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[8] John Sylak-Glassman. The Composition and Use of the Universal Morphological Feature Schema (UniMorph Schema), June 2016.