Surface-Level Morphological Segmentation of Low-resource Inuktitut Using Pre-trained Large Language Models

Mathias Stenlund[1], Hemanadhan Myneni[1], Morris Riedel[1, 2]
[1]University of Iceland, [2]Forschungszentrum Jülich
hem29@hi.is, myneni@hi.is, morris@hi.is

Abstract

Segmenting languages based on morpheme boundaries instead of relying on language independent segmenting algorithms like Byte-Pair Encoding (BPE) has shown to benefit downstream Natural Language Processing (NLP) task performance. This can however be tricky for polysynthetic languages like Inuktitut due to a high morpheme-to-word ratio and the lack of appropriately sized annotated datasets. Through our work, we display the potential of using pre-trained Large Language Models (LLMs) for surface-level morphological segmentation of Inuktitut by treating it as a binary classification task. We fine-tune on tasks derived from automatically annotated Inuktitut words written in Inuktitut syllabics. Our approach shows good potential when compared to previous neural approaches. We share our best model to encourage further studies on down stream NLP tasks for Inuktitut written in syllabics.

1 Introduction

The Inuktitut language, indigenous to the northernmost regions of Canada and spoken by roughly 40K speakers, is particularly difficult to adapt NLP tools for. Not only is the lack of appropriately sized annotated datasets a big hurdle, but so is the polysynthetic nature of the language itself. This linguistic attribute results in a very high average morpheme-to-word ratio, by some estimates as high as 4.39 (Roest et al., 2020), where often times one or two words in Inuktitut can express what would take a full sentence to express in English (Mallon, 2000) (see Figure 1). Naturally, this leads to numerous ways of forming unique and rare words, each one conveying rich linguistic information.

ᐸᕆᒧᓐᖓᐅᓂᕋᓚᐅᖅᓯᒪᓐᖏᑦᑐᖓ

Parimunngauniralauqsimanngittunga

I never said I wanted to go to Paris

Figure 1: An example of an Inuktitut word written in Inuktitut syllabics, romanized as "*Parimunngauniralauqsimanngittunga*", translating to a full sentence in English.

To combat similar issues with rare or unique words in other languages, a common practice is to pre-process textual data by deploying algorithms such as BPE (Sennrich et al., 2016) or SentencePiece (Kudo and Richardson, 2018) that are efficient at breaking up words into more digestible sub-strings. However, these algorithms are language independent and split words based on the frequency of commonly occurring sub-string character clusters and not on the basis of actual linguistic information. Instead, we turn our attention to surface-level morphological segmentation, as explicit morphological information has shown to be valuable for various down steam NLP tasks (Dyer et al., 2008; Creutz et al., 2007; Ruokolainen et al., 2016), especially for low-resource languages (Wiemerslage et al., 2022). Despite the existence of an invaluable rule-based tool (Farley, 2009) capable of segmenting Inuktitut based on linguistic information, it is not reliable as it fails to return segmentations for many words.

In this study, we explore a different approach to segmenting Inuktitut compared to previous efforts by leveraging off a pre-trained multilingual LLM and by turning surface-level morphological segmentation into a binary classification task through the use of LLMSegm (Pranjić et al., 2024). We annotate additional training data using the existing rule-based segmentation tool and evaluate our

Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 688–696
March 3-4, 2025

fine-tuned models on a variation of human annotated and automatically annotated test sets. Contrarily to the majority of previous studies, which employ the romanized version of the language, our setup focuses on segmenting Inuktitut written in Inuktitut syllabics. By sharing our best performing model, we hope to inspire others to also conduct their research on Inuktitut written in syllabics without romanizing the language first. Our main contributions are:

1. We show the potential of deploying pre-trained LLMs for surface-level morphological segmentation of Inuktitut compared to previous approaches.

2. We encourage more research to be done on down-steam NLP tasks for Inuktitut written in syllabics by making our model available[1].

2 Background and related work

There are plenty of methods dealing with morphological segmentation. Here we mention a few related to our work. Creutz and Lagus (2002) introduced an unsupervised probabilistic morpheme identifying method that has seen widespread use, with many related projects following their lead (Kohonen et al., 2010; Smit et al., 2014). More recently, Eskander et al. (2020) introduced MorphAGram, another unsupervised approach based on adaptor grammars (Johnson et al., 2006). Semi-supervised methods incorporating conditional random fields have also been proposed (Ruokolainen et al., 2014), as well as fully supervised ones (Cotterell et al., 2015). Additionally, there have been numerous neural approaches (Wang et al., 2016; Micher, 2017; Kann et al., 2018) using various model architectures. Recently, Pranjić et al. (2024) leveraged off pre-trained LLMs to segment words by turning morphological segmentation into a binary classification task. They displayed the effectiveness of their approach for a number of languages in a low-resource setting. Additionally, surface-level segmentation as a community task has also been highlighted during the 2005 to 2010 Morpho Challenges (Kurimo et al., 2010) and for a few low-resource languages in the shared task LowResourceEval-2019 (Klyachko et al., 2020).

2.1 Previous approaches for segmenting Inuktitut

The UQAILAUT Inuktitut Morphological Analyzer (Farley, 2009) is an openly available morphological analyzer for the language, developed at the National Research Council of Canada (NRC). The analyzer is a finite state transducer that makes use of hand-crafted rules to return both a surface-level morphological segmentation of an input word, and the lemma of each individual morpheme. The segmentations returned are not always unambiguous since Inuktitut words can often be correctly segmented in many ways and, consequently, for many words, more than one segmentation is returned. Unfortunately, the analyzer suffers from a flaw in that for many words, it does not return any decompositions at all, making it rather unreliable to use as a pre-processing tool for downstream tasks. In an effort to cover for words that UQAILAUT cannot process, Micher (2017) annotated more training data from the Nunavut Hansard Inuktitut-English Parallel Corpus 3.0 (Joanis et al., 2020) using the same analyzer to train a Segmental Recurrent Neural Network (SRNN) (Kong et al., 2016) for both segmentation and tagging of morpheme specific information. Le and Sadat (2020) took a different approach and deployed a bidirectional Long-Short Term Memory (LSTM) incorporating pre-trained embeddings for Inuktitut. Roest et al. (2020) trained a transformer (Vaswani et al., 2017) based model and combined it with *UQAILAUT* and BPE to form a 3-step method to segment the language. More recently, Khandagale et al. (2022) extended their adaptor grammar based tool MorphAGram with expert-based linguistic priors for morphological segmentation of Inuktitut.

3 Methodology and experimental setup

3.1 Model

For all of our experiments, we utilize Glot500-m (Imani et al., 2023), a multilingual LLM covering more than 500 languages, many of which can be considered low in resources. It builds upon the XLM-R-base multilingual model (Conneau et al., 2020) by extensively extending its vocabulary from 250K tokens to 401K, and through continued training with a masked language modelling objective. It was trained on Glot500-c[2], a

[1] Available here: https://huggingface.co/matsten/Glot500-m-iuseg

[2] Available here as a Huggingface dataset: https://huggingface.co/datasets/cis-lmu/Glot500

subset of the larger Glot2000-c corpus, containing roughly 126GB of text covering more than 400 languages, including Inuktitut. The model was evaluated on a diverse set of tasks and displayed great improvements on the newly introduced languages and also performs equal to, or better, than XLM-R-base on already seen languages.

3.2 Datasets and preprocessing

We make use of the Inuktitut side of the Nunavut Hansard Inuktitut-English Parallel Corpus 3.0[3] (Joanis et al., 2020), which contains around 8M Inuktitut words worth of debate proceedings from the Legislative Assembly of Nunavut. After running the recommended accompanying spelling normalization scripts, we extract each unique word and end up with a vocabulary of approximately 1,1M unique words, which we automatically annotate using the UQAILAUT analyzer (Farley, 2009). For each successfully analyzed word, it returns either a single or many possible surface-level morphological decompositions. Similarly to the reasoning by Micher (2017); Roest et al. (2020), we assume that words with single decompositions are the least ambiguous and therefore the most correctly labeled words. Roest et al. (2020) even show that training their transformer based segmenter on fewer amounts of unambiguous word segmentations is preferred compared to training on many ambiguous ones. We therefore follow their steps. Since Glot500-m has seen large parts of the Nunavut Hansard corpus during pre-training, we make sure that the train, validation and test/evaluation splits are divided in such a way that there are no unique words in the test/evaluation and validation split that also occur in the Glot500-c dataset. Of a total of 54,138 unambiguously segmented words, 45,231 are used for training, 3102 for validation and 3102 for test/evaluation. We refer to this test/evaluation set as the *test* set. In order to compare our approach to UQAILAUT's performance, we also evaluate our approach on another dataset separate from our initial test set. This dataset[4], referred to as the *gold* set, consists of around 1K hand-annotated Inuktitut words based on the most frequently occurring words in the Nunavut Hansard corpus. However, it is worth highlighting that, through the work of (Khandagale et al., 2022), we are made aware of the flaws of this gold set. Even though they corrected the gold set with the help of linguists and even added 100 new entries, their version of the gold set is seemingly absent from their GitHub repository.

[3] https://nrc-digital-repository.canada.ca/eng/view/object/?id=c7e34fa7-7629-43c2-bd6d-19b32bf64f60

[4] https://github.com/LowResourceLanguages/InuktitutComputing/blob/master/Inuktitut-Java/ressources/goldstandardHansard.txt

3.3 Turning segmentation into a binary classification task

We fine-tune our models on binary classification tasks derived from the annotated Inuktitut words described in Section 3.2 using LLMSegm (Pranjić et al., 2024) using the original code[5]. LLMSegm derives binary classification tasks from a word by introducing a custom morpheme boundary token, represented here as "@", that is inserted into a word between two characters. This is repeated for each unique position between two characters in the word forming $n - 1$ tasks where n is the number of characters in the word and the task is to predict whether "@" is positioned at a true morpheme boundary (see Figure 2).

Word	Tasks	Label
ᐸᕐᓇᓂᕐᒧ	ᐸ@ᕐᓇᓂᕐᒧ	0
	ᐸᕐ@ᓇᓂᕐᒧ	0
	ᐸᕐᓇ@ᓂᕐᒧ	1
	ᐸᕐᓇᓂ@ᕐᒧ	0
	ᐸᕐᓇᓂᕐ@ᒧ	1

Figure 2: Visualization of the tasks derived from an Inuktitut word, where the morpheme separator token denoted by "@" is inserted between each unique position between two characters. The task then becomes to predict whether this is a True (1) or False (0) morpheme boundary.

Additionally, prepended to each individual task is the same untouched word in its entirety, immediately followed by another custom token called the word boundary token, represented in text as "‡", effectively separating the prepended word and

[5] Original code is available here: https://github.com/sharpsy/llm-morphological-segmenter

the given task (see Figure 3).

Task	Input	Label
<@ᖅᓪᓂᖅᓗ	<ᖅᓪᓂᖅᓗ‡<@ᖅᓪᓂᖅᓗ	0

Figure 3: Visualization of the full model input for a single example prediction. Here "‡" represents the word boundary token and "@" the morpheme boundary token.

By doing this, Pranjić et al. (2024) hope to prevent the loss of information from the tokens that the pre-trained model's tokenizer normally would split the word into. By additionally including the untouched word, all original tokens are guaranteed to be retained in the input since the tokenizer will be forced to split any tokens in its vocabulary that bridges across "@". We experiment both with and without this addition by performing minimal alterations to the original code. This extra prepended word will henceforth be referred to as the *supporting word*.

3.4 Working with syllabics

We work with Inuktitut written in syllabics for two main reasons. Firstly, it is necessary since Glot500-m was fine-tuned on Inuktitut text written in syllabics. Secondly, we hypothesize that working with Inuktitut written in syllabics, as opposed to romanized Inuktitut, might be more beneficial when utilizing LLMSegm given how each input word is turned into $n - 1$ classification tasks. Since many of the syllabic characters often equate to two or sometimes even three roman characters when transcribing, the average romanized Inuktitut word often contains many more characters than the same word written in syllabic characters. Consequently, more tasks would be derived from the romanized word, which on the one hand would mean more total training samples, but among these, some might be less relevant. We say this on the basis of observations from transcription experiments[6] we do to and from syllabics. We take notice that the vast majority of morpheme boundaries in the romanized version of the language occur between characters, or clusters of characters, that would normally be transcribed into separate syllabic characters in the equivalent transcription of the same word. By working with syllabics, we thus eliminate segmentation tasks that would otherwise be derived from between roman characters that are normally represented by the same single syllabic character (see Figure 4). We deem these tasks less relevant since, according to our observations, morpheme boundaries are less likely to occur between these characters.

Word	Truth	Potential segmentations
ᓚᖅᔪᖅᑐᑦ	ᓚᖅᔪᖅ@ᑐᑦ	ᓚ@ᖅ@ᔪ@ᖅ@ᑐ@ᑦ
makkuktut	makkuk@tut	ma@k@ku@k@tu@t

Figure 4: The syllabic version of the language allows us to avoid deriving tasks such as classifying whether a morpheme boundary, denoted by "@", is present between "m" and "a", "k" and "u", and "t" and "u". This is because the character clusters "ma", "ku" and "tu" are represented as one syllabic character each, and therefore an internal boundary between them is unlikely.

This way, not only do we clear our total task pool of these hypothetically less relevant tasks, but we also create a more balanced dataset with a more evenly distributed true-to-false label ratio, as opposed to if we stick with the romanized version of the language. We calculate that out of all the tasks derived from our syllabic train set, roughly 41% are labeled as true while the rest are false. We estimate that the same train set in roman characters would have a much lower ratio of roughly 23% true labels. How effective our reasoning is will however have to be left for future efforts.

3.5 Model fine-tuning

Using the training data described in Section 3.2, we fine-tune Glot500-m for classification using LLMSegm by following the original paper (Pranjić et al., 2024). We utilize the same hyperparameters of device batch size of 256, learning rate of 2e-5, weight decay of 0.01, 20 warm up steps and AdamW optimizer (Loshchilov and Hutter, 2019). Unlike the original paper, we also fine-tune a second model without the supporting word to investigate how this affects training and later performance. For each fine-tuning set up, we train 10 separate models on randomly sampled variations of the original training data (with replacement) and pick the best performing one for evaluation.

[6]We transcribe using Yudit: `https://yudit.org/`

We call the model trained without the supporting word *Glot500-m-iuseg-n* and the one with the supporting word *Glot500-m-iuseg-s*. All model training is done using 4x Nvidia A100 GPUs.

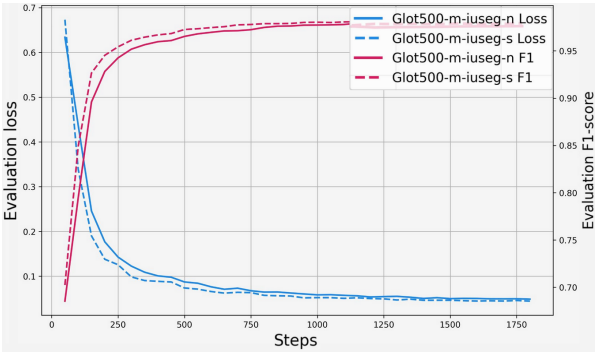

Figure 5: The evaluation loss and F1-score for (*Glot500-m-iuseg-n*) and (*Glot500-m-iuseg-s*).

During training, we take notice that the model training with the supporting word improves slightly faster than the model training without, both in terms of evaluation loss and evaluation F1-score (see Figure 5). This suggests that the tokens of the original unsegmented word produced by the tokenizer might indeed help retain valuable linguistic information from the pre-training that aids the fine-tuning process.

3.6 Evaluation

We evaluate our models on the two evaluation sets described in Section 3.2 (test and gold) and report back F1-score based on the difference between predicted morpheme boundaries and the actual boundaries. Much like (Kann et al., 2018; Roest et al., 2020; Pranjić et al., 2024), we additionally complement our F1-score by reporting the accuracy score calculated as the proportion of all words where every morpheme boundary was correctly predicted. We then end up with two complementary metrics, one calculated at morpheme-level and one at word-level. For comparison, we treat the Glot500-m (Imani et al., 2023) tokenizer as our baseline and also compare our results to previous studies where it is applicable. Due to the UQAILAUT analyzer's tendency to fail when presented with certain words, we also evaluate a combined custom setup where our best performing model processes these failed words. We call this setup UQAILAUT+.

4 Results & discussion

We present our results in Table 1 and compare where possible to the following: *AG-SS* (Khandagale et al., 2022), *Trf. (45K single)* and *3-step* (Roest et al., 2020), *LSTM* with pre-trained embeddings (Le and Sadat, 2020), *SRNN CG* (Micher, 2017) and *UQAILAUT* (Farley, 2009). Our fine-tuned models *Glot500-m-iuseg-n* and *Glot500-m-iuseg-s* show the potential of our chosen methods compared to previous neural approaches in terms of F1-score and accuracy. Both of our models achieve a worse accuracy on the gold set, albeit higher F1, compared to the *3-step* setup.

Model/setup	**Test**		**Gold**	
	F1	**Acc.**	**F1**	**Acc.**
Glot500-m tok.	0.59	0.04	0.42	0.18
AG-SS	-	-	0.60*	-
Trf. (45K single)	-	-	0.68	0.54
3-Step	-	-	0.74	0.70
LSTM	0.75*	-	-	-
SRNN CG	0.95*	-	-	-
Glot500-m-iuseg-n	**0.98**	0.89	0.85	0.61
Glot500-m-iuseg-s	**0.98**	**0.90**	0.87	0.66
UQAILAUT	-	-	0.92	0.74
UQAILAUT+	-	-	**0.95**	**0.81**

Table 1: F1-score and accuracy scores from our models compared to previous studies. "-" indicates that evaluation metrics for the particular dataset were never reported or that they can not be reported. "*" next to a score indicated that the score was reported on a variation of the same dataset compared to what was used for evaluation in this study.

Worth noting is that where Micher (2017) choose 1K unambiguous samples annotated by UQAILAUT as their test set and Le and Sadat (2020) use 250 sentences as their test, we select as many unambiguous samples as possible who's exact word form does not also appear in the training data of the Glot500-m model for a total of 3102. Hence, they are all evaluated on different amounts of words, and most likely also different words, from the Nunavut Hansard corpus (Joanis et al., 2020). Our model *Glot500-m-iuseg-n* slightly underperforms *Glot500-m-iuseg-s* trained using the supporting word. This would suggest that there is some benefit to including the supporting word not

only during training, but also during evaluation, possibly due to a retention of information from pre-training. This is also implied to be the case, since the Glot500-m tokenizer's decent F1-score hints to the existence of some underlying knowledge of how to segment Inuktitut words, despite it returning very few fully correctly segmented words. None of the neural approaches alone outperform the UQAILAUT analyzer in terms of F1-score and accuracy, even though *Glot500-m-iuseg-s* is close. The combined setup UQAILAUT+ however, achieves the highest score on the gold set. Even though this setup does not improve F1-score too much, it improves accuracy by a not insignificant amount.

4.1 Oversegmentation

When examining the predictions on the two evaluation sets by our best performing model, we take notice of its tendency to oversegment words containing fewer than 4 true segmentations, peaking at words with 0 (see Figure 6). Going from 4 to 8 true segmentations per word, our best model achieves a more stable predicted-segmentations-per-word to true-segmentations-ratio on the test set, but seemingly underpredicts on the gold set for words in the same range.

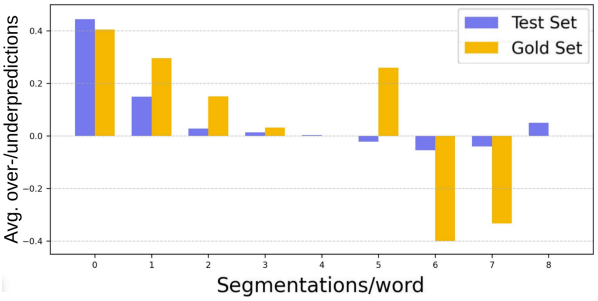

Figure 6: The average amount of morpheme boundaries over-/under predicted by *Glot500-m-iuseg-s* (y-axis) for words with n true segmentations from the test and gold set (x-axis).

Additionally, by calculating isolated F1-scores on predictions for words with 0-1 true segmentations, we see that our model performs much worse in this range compared to F1-scores in all the other ranges (see Figure 7). This underperformance is also reflected in the drop in F1-score between evaluations on the test set and the gold set, going from 0.98 to 0.87, since the gold set is made up of around 60% words in the range of 0–1 segmentations per word. The fact that our model saw many more words with segmentations in the range of 2-5 compared to the range 0-1 during fine-tuning might help explain why our model performs worse for these words. In fact, the average number of segmentations per word in our train set is much higher than in the gold set, as displayed in Table 2.

average	**train**	**test**	**gold**
seg./word	3.3	3.5	1.6
char./word	9.2	9.7	6.3

Table 2: Average true segmentations and syllabic characters per word in the train, test and gold set.

This suspicion is also supported by the higher F1-scores for words with true segmentations ranging from 2-5. Further building on this argument, the way the LLMSegm tool turns each annotated word into $n-1$ segmentation tasks amplifies this training imbalance, as words with fewer segmentations typically contain fewer characters. This means that our model will see longer words many more times compared to shorter words. For this reason, we try to mitigate this imbalance by fine-tuning additional models where we upsample words in the segmentation-per-word range of 0-1 by 2x and 3x in the training data but with no positive effect on performance.

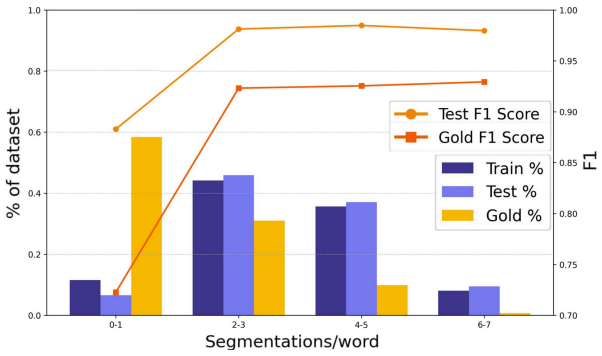

Figure 7: The percentage of words in the dataset that contain a certain amount of segmentations per word, as well as F1-score performance of *Glot500-m-iuseg-s* on words in each individual bracket for both the test and gold set.

Ignoring our model's struggle with shorter words, we have two possible explanations for why our models perform worse overall on the hand an-

notated gold set than on the test set. Since both the training data and our test set were automatically annotated by UQAILAUT that itself does not score perfectly on the gold set, we can only assume that some of the training and test data were also incorrectly annotated. This might have inflated the scores on the test set compared to the gold set, and might also mean that our model will make the same mistakes when deployed as a pre-processing tool. We also know that the gold set contains 1K of the most frequent words in the Nunavut Hansard corpus, while our model was fine-tuned on unique words where word frequency was not taken into consideration.

4.2 UQAILAUT issues

As mentioned previously, the UQAILAUT analyzer is unable to produce decompositions for many Inuktitut words. This is despite it outperforming all other setups. We are unsure of the exact cause of the UQAILAUT analyzer's inability to process certain words, but a quick look at these failure cases suggest that it might have to do with spelling inconsistencies and or not enough coverage in its hand-crafted rules to account for these. This might in turn explain why in the UQAILAUT+ setup, our model was able to correctly process a few words where UQAILAUT fails since spelling inconsistencies do not automatically result in a failed attempt thanks to the more dynamic nature of our neural model. However, due to the small evaluation dataset, it is not possible to draw any definitive conclusions.

When evaluating only the UQAILAUT analyzer on the gold set, we take notice that it fails to return any decompositions at all for approximately 11% of the words. However, when annotating the unique words from the Nunavut Hansard corpus to create our dataset, we note that, much like the observations made by Micher (2017), this percentage increases to approximately 30%. This suggests that, despite its high scores on the gold set, UQAILAUT is unfit to pre-process real world texts for downstream NLP tasks on its own since some very long words would be left unsegmented. Further, this suggests a performance decrease in a scenario where we have access to more human annotated gold data for evaluation that contains rarer words and not just the 1K most common ones. In fact, we calculate that only 20% of all word forms in the Nunavut Hansard corpus occur more than once and only 11% more than twice. This abundance of unique words in Inuktitut further highlights the importance of continued research in the field to ultimately benefit downstream NLP tasks.

5 Conclusion

We contribute to ongoing research focusing on the polysynthetic language Inuktitut by fine-tuning and sharing a Glot500-m LLM for binary classification of morpheme boundaries. Our best model shows promising results when comparing to previous efforts, despite struggling to segment words with fewer true segmentation boundaries. We also show the potential of deploying existing pre-trained LLMs using LLMSegm even for under-resources polysynthetics languages without the need to train anything from scratch. Additionally, we further encourage future studies on down stream NLP tasks for Inuktitut written in syllabics. In future efforts, we intend to improve the performance of our model, as well as investigate its potential as a pre-processing tool for down stream NLP tasks such as machine translation.

6 Limitations

The main limitation with LLMSegm is the fact that it completely relies on the existence of a pre-trained model that has seen the target language during pre-training, which, ironically, excludes many of the world's lowest resource languages. Additionally, being a low-resource language, Inuktitut suffers from a lack of well-balanced human segmented gold data for both training and evaluation. Thus, it is not possible to draw solid conclusions based on evaluation on the only available gold set, and only further highlights the need for more such data. Our method also does not take alternative segmentations into consideration, but we still believe that our model can be used as a pre-processing tool to benefit down stream performance. Further, the accuracy, as reported by Roest et al. (2020), Pranjić et al. (2024), and now also by us, is not an ideal metric for evaluating a segmenter for polysynthetic languages. Since this definition of accuracy gives the same weight to words containing different amounts of segmentations, a correctly predicted decomposition of a word containing 1 true segmentation is valued higher than a word containing 8 true segmentations, where the setup only successfully predicts 7.

Acknowledgments

This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 101135671 (TrustLLM). It is co-financed by the EU-ROCC2 project, funded by the European High-Performance Computing Joint Undertaking (JU) and EU/EEA states under grant agreement No 101101903. Parts of this were also supported by the European Digital Innovation Hub (EDIH) of Iceland (EDIH-IS), partially funded by the Digital Europe Programme. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer JUWELS at the Jülich Supercomputing Centre (JSC).

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ryan Cotterell, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-Markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164–174, Beijing, China. Association for Computational Linguistics.

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Trans. Speech Lang. Process.*, 5(1).

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, MPL '02, page 21–30, USA. Association for Computational Linguistics.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio. Association for Computational Linguistics.

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith Klavans, and Smaranda Muresan. 2020. MorphAGram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7112–7122, Marseille, France. European Language Resources Association.

Benoît Farley. 2009. The uqailaut project. `https://www.inuktitutcomputing.ca/index.php`. Accessed: 2024-07-15.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2006. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza-Ruiz, and Hinrich Schütze. 2018. Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 47–57, New Orleans, Louisiana. Association for Computational Linguistics.

Sujay Khandagale, Yoann Léveillé, Samuel Miller, Derek Pham, Ramy Eskander, Cass Lowry, Richard Compton, Judith Klavans, Maria Polinsky, and Smaranda Muresan. 2022. Towards unsupervised morphological analysis of polysynthetic languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 334–340, Online only. Association for Computational Linguistics.

Elena Klyachko, Alexey Sorokin, Natalia Krizhanovskaya, Andrew Krizhanovsky, and Galina Ryazanskaya. 2020. Lowresourceeval-2019: a shared task on morphological analysis for low-resource languages.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.

Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. Segmental recurrent neural networks.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95, Uppsala, Sweden. Association for Computational Linguistics.

Tan Ngoc Le and Fatiha Sadat. 2020. Low-resource NMT: an empirical study on the effect of rich morphological word segmentation on Inuktitut. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 165–172, Virtual. Association for Machine Translation in the Americas.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Mick Mallon. 2000. Inuktitut linguistics for technocrats. `https://www.inuktitutcomputing.ca/Technocrats/ILFT.php`. Accessed: 2024-07-15.

Jeffrey Micher. 2017. Improving coverage of an Inuktitut morphological analyzer using a segmental recurrent neural network. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 101–106, Honolulu. Association for Computational Linguistics.

Marko Pranjić, Marko Robnik-Šikonja, and Senja Pollak. 2024. LLMSegm: Surface-level morphological segmentation using large language model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10665–10674, Torino, Italia. ELRA and ICCL.

Christian Roest, Lukas Edman, Gosse Minnema, Kevin Kelly, Jennifer Spenader, and Antonio Toral. 2020. Machine translation for English–Inuktitut with segmentation, data acquisition and pre-training. In *Proceedings of the Fifth Conference on Machine Translation*, pages 274–281, Online. Association for Computational Linguistics.

Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. 2016. A comparative study of minimally supervised morphological segmentation. *Computational Linguistics*, 42(1):91–120.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2014. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 84–89, Gothenburg, Sweden. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Linlin Wang, Zhu Cao, Yu Xia, and Gerard de Melo. 2016. Morphological segmentation with window lstm neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30.

Adam Wiemerslage, Miikka Silfverberg, Changbing Yang, Arya McCarthy, Garrett Nicolai, Eliana Colunga, and Katharina Kann. 2022. Morphological processing of low-resource languages: Where we are and what's next. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 988–1007, Dublin, Ireland. Association for Computational Linguistics.